09-05-07

OIPE IAP20 SEP 04 2007 PATENT & TRADEMARK OFFICE

Attorney's Docket No. 042933/298965

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Nativadade Lobo — Confirmation No.: 5615
Appl. No.: 09/625,201
Filed: July 21, 2000
For: PULSE SHAPING WHICH COMPENSATES FOR COMPONENT DISTORTION

BOX ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMITTAL OF PRIORITY DOCUMENT

To complete the requirements of 35 U.S.C. § 119, enclosed is a certified copy of Great Britain priority Application No. 9814300.1, filed July 1, 1998.

Respectfully submitted,

Cory C. Davis
Registration No. 59,932

Customer No. 00826
Alston & Bird LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Charlotte Office (704) 444-1000
Fax Charlotte Office (704) 444-1111

"Express Mail" mailing label number EV521113148US
Date of Deposit September 4, 2007

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: BOX ISSUE FEE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Tamara Stevens

Intellectual
Property
Office

reativity and Innovation

Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with patent application GB9814300.1 filed on 1 July 1998.

I also certify that the attached copy of the request for grant of a Patent (Form 1/77) bears an amendment, effected by this office, following a request by the applicant and agreed to by the Comptroller-General.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-egistration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the ompany in this certificate and any accompanying documents shall be treated as references to e name with which it is so re-registered.

accordance with the rules, the words "public limited company" may be replaced by p.l.c., c, P.L.C. or PLC.

under the Companies Act does not constitute a new legal entity but merely any to certain additional company law rules.

Signed

Dated 21 August 2007

Patents Form 1/77

Patents Act 1977
(Rule 16)

The Patent Office

1/77

THE PATENT OFFICE
1 - JUL 1998
RECEIVED BY HAND

02JUL98 E372674-1 D02716
P01/7700 25.00 - 9814300.1

THE PATENT OFFICE
1 JUL 1998
RECEIVED BY HAND

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

01 JUL 1998

1.	Your reference	PAT 98014 GB
2.	Patent application number *(The Patent Office will fill in this part)*	**9814300.1**
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	NOKIA MOBILE PHONES LIMITED KEILALAHDENTIE 4 02150 ESPOO FINLAND
	Patents ADP number *(if you know it)*	5911995004
	If the applicant is a corporate body, give the country/state of its incorporation	
4.	Title of the invention	RADIO TELEPHONE
5.	Name of your agent *(if you have one)* "Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*	~~HELEN L HAWS (MRS)~~ ~~NOKIA MOBILE PHONES~~ ~~PATENT DEPARTMENT~~ ~~ST GEORGES COURT~~ ~~ST GEORGES ROAD~~ ~~CAMBERLEY~~ SURREY GU15 3QZ NOKIA R&D (UK) LTD, Patent Department, Ashwood House, Pembroke Broadway, CAMBERLEY, Surrey, GU15 3XD
	Patents ADP number *(if you know it)*	4105177001 A JW 20/11/98

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*
GB	9801302.2	21.01.98
GB	9801305.5	21.01.98
GB	9801306.3	21.01.98

SEE ATTACHED SHEET

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
   a) any applicant named in part 3 is not an inventor, or
   b) there is an inventor who is not named as an applicant, or
   c) any named applicant is a corporate body.
   See note (d))

YES

I certify this to be a true copy.
G D Gort
Acting for Comptroller

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	33 pages + 24 pages ANNEX 1
Claim(s)	7
Abstract	1
Drawing(s)	13

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	1
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature [signature] Date 1st July 1998

HELEN L HAWS (MRS) - Agent for the Applicant

12. Name and daytime telephone number of person to contact in the United Kingdom — KENDRA JEFFERY - 01276 419538

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

OUR REF: PAT 98014 GB

APPLICANT: NOKIA MOBILE PHONES LIMITED

TITLE: RADIO TELEPHONE

COUNTRY	PRIORITY APPLICATION NUMBER	DATE OF FILING
GB	9801308.9	21.01.1998
GB	9804600.6	05.03.1998
GB	9805126.1	11.03.1998
GB	9805234.3	12.03.1998
GB	9805504.9	13.03.1998

Radio Telephone

In digital radio telephones serial bit streams of data are transmitted by modulating a carrier. There are several types of modulation schemes used to transmit bit streams. One common performance criterion in the design of a modulation scheme is bandwidth efficiency, defined as the ratio of data rate to channel bandwidth.

Such modulation schemes may be linear or non-linear. Linear schemes are considered to be those which follow the general rule that $f(\propto x) = \propto f(x)$, whereas non-linear schemes do not follow this rule, usually by virtue of their complex nature. Examples of non-linear modulation schemes include most phase modulation schemes which do not meet the aforementioned linearity rule, such as GMSK. Examples of linear modulation schemes include amplitude modulation and phase modulation schemes which meet the linearity rule, such as QPSK. Whilst QPSK is a form of phase modulation consisting of complex parts, it can be considered as linear because the two bits used to form each symbol independently modulate the in phase and quadrature phase channels at the baseband of the signal. To date, TDMA telecommunication systems have used both linear and non-linear modulation schemes (for example GSM uses the non-linear GMSK scheme whereas PDC uses the linear QPSK scheme) whereas CDMA telecommunication systems have only used linear modulation schemes (for example QPSK). The existing partnership between telecommunication systems and modulation schemes provide certain advantages and disadvantages. These are discussed below in relation to GSM/GMSK and CDMA/QPSK.

GMSK is a phase modulation that converts a serial bit stream into a phase shift of a carrier wave. The function of the modulation is to convert the incoming serial bit stream into analog signals that modulate the carrier of the transmitter. In GMSK the outgoing phase shift is filtered. The Gaussian function acts as a filter, removing the sharp edges of the digital pulses. Without this filtering the required bandwidth to transmit the signal would be far greater. Even with the gaussian filter it is

acknowledged that the GSM system is spectrally inefficient. For example with a BT product of 0.3 GSM has a bandwidth efficiency lower than that of QPSK with $\propto = 0.22$. The GMSK modulation does, however, provide a constant amplitude signal that is power efficient. In theory, the signal for example, does not get distorted when amplified by a non-linear PA.

In existing CDMA systems a different phase modulation technique, QPSK, is chosen to provide a higher bit rate or better bandwidth efficiency than the GMSK. In QPSK orthogonal signals are transmitted which double the data rate relative to MSK modulation. In QPSK modulation the outgoing phase shift is Nyquist filtered to provide root raised cosine shaped pulses that increase the bandwidth efficiency and reduced bit error rate by eliminating intersymbol interference. Although QPSK with root raised cosine pulse shaping is spectrally efficient allowing a high data rate and providing a low BER, it needs a linear PA to avoid distortion when transmitted.

The GSM system and existing CDMA systems were designed to meet user needs considered appropriate at their conception. As schemes for third generation systems are being planned the criteria desirable in a telecommunication system that will need to provide for user's needs well into the twenty-first century are being considered.

In third generation systems more bandwidth efficient modulation schemes than the second generation schemes are being considered. It will be important that the data rates are high enough to allow the expansion of the telecommunication industry from voice into data applications to continue without reducing the power efficiency below that acceptable for a battery powered terminal. None of the existing modulation schemes allow the data rates to be high enough to support the myriad of data applications that are required without sacrificing bit error rate and /or power amplifier efficiency to an unacceptable extent.

In the present invention it has been realised that non-linear modulation schemes can be used in spread spectrum telecommunication systems. Such as arrangement

ovides the high power amplifier efficiency of a non-linear modulation schemes, such as GMSK, with the high user rate of a spread spectrum system, such as CDMA. This realisation goes against the general teaching in the art, in which current spread spectrum systems always employ linear modulation schemes, and in which it is taught that such a combination would result in a too complicated system.

According to one aspect, the present invention provides a relatively straight forward method of despreading a signal which has been spread and then transmitted by a non-linear modulation scheme. This method comprises transforming the spreading code into a succession of phasor positions and correlating the spread signal with the transformed code.

Transformation may be carried out just, for example, assuming the received bit is positive. However, as the modulation is non-linear a different transformation of the code is required when the received bit is assumed negative. Therefore, preferably transformations are carried out both assuming that the received signal is positive and negative as this eliminates the need for a complicated demodulator (ie. no viterbi or training sequence is required as is the case for GSM employing GMSK). Instead, demodulation is provided by the largest of the transformed signals once correlated.

The method may involve despreading a signal constructed using the superposition of N amplitude modulation pulses. The use of a plurality of pulses provides more energy at the receiver, so there is a better integrity of what is being sent. The pulses may be the first N pulses which approximate a Gaussian pulse shape according to Laurent's superposition theory. Alternatively, they may be pulses which are optimally shaped for a system depending on its cost function requirements (eg bit error rate, bandwidth, amplitude, AFC). The signal may, for example, be constructed using the superposition of 2 pulses (N=2) and the despreader may perform the transforming and correlating steps for each of these pulses (M=2). However, preferably even more pulses are used to construct the signal for transmission to further improve its integrity. In this event, the despreader may only

perform the transforming and correlation steps for the first M pulses, where $M<N$, reduce the processing required by the receiver without significant energy loss. For example, in a preferred embodiment M=2 and N=4.

According to another aspect of the present invention, there is provided a method of transmitting and receiving a signal in a spread spectrum telecommunications system, the method comprising:

spreading a signal to be transmitted by a code;
modulating the spread signal using a non-linear modulation scheme;
transmitting the modulated signal;
receiving the modulated signal;
transforming the code into a succession of phasor positions; and
correlating the received signal with the transformed code.

According to a further aspect of the present invention, there is provided a despreader for despreading a signal which has been spread by a code and modulated according to a non-linear modulation scheme, the despreader comprising:

means for providing the code transformed into a succession of phasor positions: and

a correlator for correlating the spread signal with the transformed code.

According to a still further aspect of the present invention, there is provided a demodulator for demodulating a signal which has been spread by a code and modulated according to a non-linear modulation scheme, the receiver comprising:

such a first despreader for despreading the received signal assuming the received bit is positive;

such a second despreader for despreading the received signal assuming the received bit is negative; and

a comparator for comparing the correlated signals output by the first and second despreaders to determine the sign of the received signal.

Also, there is provided a receiver for a communication device comprising a despreader or a demodulator according to the present invention.

Furthermore, there is provided a transceiver for a communication device comprising such a receiver and a transmitter having a spreader for spreading a signal by a code, a non-linear modulator for modulating the spread signal and means for transmitting the modulated spread signal.

Moreover, there is provided a communication device operable in a communication system, comprising such a transceiver.

According to yet another aspect of the present invention, there is provided a dual mode receiver operable in a first mode in a spread spectrum telecommunications system and a second mode in a telecommunications system which uses a non-linear modulation scheme, the receiver comprising:

a first demodulator for demodulating a received signal in accordance with the non-linear modulation scheme in the first mode of operation, comprising a despreader having means for providing the code transformed into a succession of phasor positions, and a correlator for correlating the received signal with the transformed code; and

second receiver means comprising a demodulator for demodulating a received signal in accordance with the non-linear modulation scheme in the second mode of operation.

Similarly, there is provided a dual mode transmitter operable in a first mode in a spread spectrum telecommunications system and a second mode in a telecommunications system which uses a non-linear modulation scheme, the device comprising a modulator for modulating a data signal with a carrier signal in accordance with the non-linear modulation scheme in both the first and second modes of operation and means for spreading the data signal prior to modulation in the first mode.

These dual mode receiver and transmitter may form part of a dual mode communication device.

The invention provides a dual mode device which comprises a single modulator, and thus is cheaper and reduced in size. Other components common to both modes may include correlators and frequency down converters in the receiver.

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, of which:

figure 1 is a known system CDMA;

figure 2 is a CDMA system according to an embodiment of the present invention;

figure 3 is a CDMA transmitter according to an embodiment of the present invention;

figure 4 is a CDMA receiver according to an embodiment of the present invention;

figure 5(a) is a CDMA receiver according to a preferred embodiment of the present invention;

figure 5(b) is a CDMA receiver demodulator stage according to a preferred embodiment of the present invention;

figure 6 is a dual mode GSM/CDMA transmitter according to an embodiment of the present invention;

figure 7 is a dual mode GSM/CDMA receiver according to an embodiment of the present invention;

figure 8(a) shows an example of a received signal for a spread +1 bit;

jure 8(b) shows transformed Gold Code for detecting a +1 bit according to an embodiment of the present invention;

figure 9(a) shows an example of a received signal for a spread –1 bit;

figure 9(b) shows transformed Gold Code for detecting a –1 bit according to an embodiment of the invention;

figure 10a is a graph of amplitude against time of a +1 bit having been despread by a receiver arranged to detect a +1 bit;

figures 10b, 10c and 10d are a graph of real, imaginary and absolute values respectively of amplitude against time of a +1 bit having been despread by a receiver arranged to detect a +1 bit, emphasising the difference between the desired signal and noise by showing values for only 20 instants;

figure 11 is a graph of amplitude against time of a signal having been despread by a receiver using an incompatible code;

figure 12a is a graph of amplitude against time of a -1 bit having been despread by a receiver arranged to detect a -1 bit;

figure 12b is a graph of amplitude against time of a -1 bit having been despread by a receiver arranged to detect a +1 bit;

figure 13a and 13b are graphs of amplitude against time of a despread first pulse of the received signal, using different transformation methods; and

figures 13c and 13d are graphs of amplitude against time of a despread second pulse of the received signal, using different transformation methods.

In spread spectrum systems unique digital code, rather than separate RF frequencies or channels are used to differentiate mobile stations. The codes are

shared by both the mobile stations and base stations and are called pseudorandom binary codes, one type of which is gold code. Figure 1 illustrates an example of such a spread spectrum system, known as a direct sequence system, which utilises a linear modulation scheme.

In figure 1, information A is modulated on to a carrier B. The modulated signal C is then modulated by a pseudorandom binary code, gold code D, which spreads the signal in the frequency domain. At the receiver, the incoming signal is down converted to a suitable intermediate frequency and multiplied with a synchronised encoded replica F of the gold code which is specific to that receiver, effectively despreading the received signal. The resultant signal G can then be conventionally demodulated to extract the information H.

Assuming base station transmitter 11 is provided with information for both mobile station receiver A, referenced 12 and mobile station receiver B, referenced 13, then the encoded signal E transmitted by the base station transmitter 11 can be represented by:

$E = En(a,info_a) + En(b,info_b)$

where En = encoded signal

a,b = Gold code for receiver A (12), receiver B (13)

$info_a$, $info_b$ = information for receiver A (12), information for receiver B (13)

This signal is decoded by the receivers A and B.

At receiver A,

$G = De(a,(En(a,info_a) + En(b,info_b)))$

Where De = decoded process

Because a linear modulation scheme is used, these components can be separated so that

$G = De(a,En(a,info_a)) + De(a,En(b,info_b)$

$= info_a + \eta_b$

Where η= noise

Similarly, at receiver B,

K = De(b,(En(a,info$_a$) + En(b,info$_b$)))

$\equiv$ De(b,En(a,info$_a$)) + De(b,En(info$_b$)

$= \eta_a$ + info$_b$

The spreading and despreading of an information signal occurs as follows.

Consider that the information to be transmitted is for receiver A and consists of 4 bits A = info$_a$ = {-1,1,-1,1} and the Gold Code a is a 255 bit Gold Code.

Then, the spreading modulator outputs signal E having 4 x 255 bits.

ie. $(-1a)\bigcup(1a)\bigcup(-1a)\bigcup(1a)$

Despreading modulator of mobile station receiver A then takes the first 255 bits of received signal E and performs a dot product operation (correlation) with Gold Code a. i.e. for the first bit of info$_a$: -1(255)=-255

The sign of the first bit of resultant signal G gives the sign of the bit transmitted. Hence, in this instance, the demodulator knows a - 1 bit was sent. The operation is then carried out for the second 255 bits of the received signal, and so on, until the received signal is completely demodulated.

Examples of simple encoding/decoding

For a + 1 information bit, assuming a 5 bit Gold Code of {-1-111-1}

Encoded signal {-1-111-1}

Decoded signal = dot product of Gold Code and encoded signal

= {11111} = 5

For a - 1 information bit, assuming the same Gold Code.

Encoded signal {11-1-11}

Decoded signal {-1-1-1-1-1} = -5

Hence, the decoded signal indicates which sign the information bit is. This is as a consequence of the system being linear, and would not be the case in a non-linear modulation scheme.

Figure 2 shows a spread spectrum system according to an embodiment of the present invention which provides a technique for decoding an encoded signal which has been transmitted using a non-linear modulation scheme, and for which the sign of the bit would not be able to be readily determined.

The base station transmitter 21 is the same as that of Figure 1, with one exception. That is, the spread signal E' is modulated using a non-linear modulation technique. The receivers 22, 23 likewise differ in that they receive a signal E' modulated using a non-linear modulation technique. Moreover the despreading modulator despreads the received signal E' by mixing it with transformed Gold Code F', J' specific to that receiver. The resultant signal G', K' is then demodulated to extract the information H', L'.

Transformation by a receiver 22, 23 of its Gold Code is based upon the following principle.

1. Consider what the constellation (phasor) diagram of the received signal E' would look like assuming the information content was a + 1 bit and assuming a starting point at quadrant zero

2. Reverse or 'undo' this signal to determine the transformed Gold Code F' which will give a high value when correlated with the received signal, to indicate detection of a +1 bit. Such a code has real parts of the same sign as the Gold Code and imaginary parts of opposite sign. (This is further exemplified below).

(Optional) Repeat steps (1) and (2) to detect a –1 bit.

An example of this principle is given below, with reference to Figures 8 and 9.

Assume that the mobile station receiver A, referenced 22 has a Gold sequence D {0,0,1,1,0}. Then if the information C is a +1 bit, the received signal E′ will be {0,0,1,1,0} whereas if the information C is a –1 bit, then the received signal E′ will be {1,1,0,0,1}.

1. Transformation for detecting a +1 bit

   Detecting a +1 bit can be shown by taking the received signal for a +1 bit E′ = {0,0,1,1,0}.

   Step 1

   This signal moves around the constellation diagram in $\frac{\Pi}{2}$ steps according to the sequence {-1,-1,1,1,-1} (where 0 → -1), as shown in Figure 8a. This can be reflected in terms of real and imaginary parts as {i –1 i 1 i}

   Step 2 – Determine transformation

   Figure 8b shows a constellation diagram of the transformed Gold Code which would give a large value for the +1 bit received signal of Figure 8(a). This can be reflected in terms of real and imaginary parts as {i –1 -i 1 -i }.

   Correlating the received signal and the transformed Gold Code gives:

   $-i^2$ 1 $-i^2$ 1 $-i^2$ = 1,1,1,1,1

   = 5 (high value)

2. Transformation for detecting a – 1 bit

   Detecting a – 1 bit can be shown by taking the received signal for a – 1 bit E′ = {1,1,0,0,1}

   and mapping it on a constellation diagram as shown in Figure 9a. (Step 1) This

can be reflected in terms of real and imaginary parts as {-i –1 –i 1 –i}.

Step 2 – Determine transformation

Figure 9b shows the reverse or 'undone' phases which form the transformed Gold Code to give a large value for the –1 bit received signal of Figure 9(a). This can be reflected in terms of real and imaginary parts as {i –1 i 1 i }.

Correlating the received signal and this transformed Gold Code gives :

$-i^2\ 1\ -i^2\ 1\ -i^2 = \underline{5}$ (high value).

Now we know suitable transformations F' which can be used by mobile station receiver A, 22, for detecting +1 bit and –1 bit. (Figs 8b and 9b respectively).

3. Example of despreading by mobile station receiver A, using these transformations

Assume that the spreading modulator of the base station transmitter 21 mixes a 2 bit information sequence C {1, -1} with the Gold Code Sequence D {0,0,1,1,0} of mobile station receiver A. The signal E' transmitted by the mobile station transmitter 21 will be C X D. i.e. {0,0,1,1,0} {1,1,0,0,1}

In one embodiment, the despreading modulator will correlate the received signal, one codes worth at a time, with the transformed Gold Code for detecting a + 1 and with the transformed Gold Code for detecting a – 1. Then, the demodulator compares the resultant correlated signals for that codes worth, and determines the sign of the bit transmitted as that which gives the highest value. This is then repeated for the next codes worth.

Taking the first code's worth of the received signal {0,0,1,1,0} (Figure 8(a)), and correlating it with the transformed Gold Code for detecting a + 1 (Figure 8(b)) gives a despread signal (product) G' of:

{i –1 i 1 i} {-i –1 –i 1 –i} = {11111} = <u>5</u>

On the other hand, taking this portion of the received signal and correlating it with the transformed Gold Code for detecting a – 1 (Figure 9(b)) gives a despread signal product G' of:

{i –1 i 1 i} {i –1 i 1 i} = {-1 1 –1 1 -1} = <u>-1</u>

The highest value indicates detection of a + 1 bit, and the demodulator demodulates this signal accordingly.

Now, taking the second code's worth of the received signal {1, 1, 0, 0, 1} (Figure 9(a)) and correlating it with the transformed Gold Code for detecting a + 1 (Figure 8(b)) gives a despread signal product G' of:

{-i –1 -i 1 -i} {-i –1 -i 1 -i} = {-1 1 –1 1 -1} = <u>-1</u>

On the other hand, taking this portion of the received signal and correlating it with the transformed Gold Code for detecting a –1 (Figure 9(b)) gives a despread signal product G' of:

{-i –1 -i 1 -i} {i –1 i 1 i} = {1 1 1 1 1} = <u>5</u>

The highest value indicates detection of a –1 bit, and the demodulator demodulates this signal accordingly.

Examples of algorithms providing transformations such as those shown in Figures 8(b) and 9(b) are outlined below. They involve the use of Laurent's superposition theory that phase modulations can be approximated by the superposition of AM pulses (Co, C_1 etc), and in this embodiment only the first of the AM pulses (Co) is considered. Laurent uses binary representations ai to define the value of the complex phase coefficient bi associated with i^{th} component. These are defined as follows, depending n whether a +1 or a –1 is to be detected:

$a_i = 1$ if $C_i = 1$ for $i = 0, \ldots N-1\}$ for detecting +1
$a_i = -1$ if $C_i = 0$ for $i = 0, \ldots N-1\}$

$a_i = -1$ if $C_i = 1$ for $i = 0, \ldots N-1\}$ for detecting -1

$a_i = 1$ if $C_i = 0$ for $I = 0, \ldots N-1\}$

where $C_i = \{C_o, C_1, \ldots C_{N-1}\}$ and N is the number of elements in the sequence.

The complex phase coefficient b_i is defined by:

$b_i = b_{i-1} + a_i$ for $i = 1, 2 \ldots N-1$

Where $b_0 = a_0$

And Laurent's approximation for the phase modulated signal is given by i^{b_i} i.e. the received signal E' in figure 2 is given by i^{b_i}.

As mentioned above, a transformation which is a reversal of the $\frac{\Pi}{2}$ phase shifts of the code gives a high value when correlated with the received signal, if that received signal is of the same sign as that associated with the transformation. Hence a transformation d_i which may be used is i^{-b_i}.

An alternative transformation which can be used is $d_i = y_i\, i^{b_i}$, where $y_i = (-1)^i$ for $i=0, 1 \ldots N-1$. This transformation is computationally efficient

If necessary, further transformations can be used in connection with subsequent pulses (C_1 etc). This is exemplified in Figure 5(b) and discussed below. The use of two or more pulses provides more energy at the receiver so there is a better integrity of what is being sent. Preferably four pulses are used.

Also, a transformation for detecting a +1 bit alone may be used. However, preferably it is carried out for detecting both ± 1 bits, as this eliminates the need for a complicated demodulator. Instead, the demodulation is merely a comparison, to

see which transformation yields the largest value when correlated with the received signal.

Figure 3 illustrates a Code Division Multiple Access (CDMA) transmitter according to an embodiment of the invention. CDMA conventionally comprises a frame made up of a dedicated physical data channel (DPDCH) and a dedicated physical control channel (DPCCH). A bit sequence 301 to be transmitted is input to a frame builder 302 of the transmitter, which puts the bits in the appropriate part of the frame (i.e. in the DPDCH).

The bit stream is then spread across the spectrum by the Gold Code Encoder. This Gold Code Encoder 303 operates as follows.

Given $\{c_0, c_1 \ldots c_{N-1}\}$	bit stream
and $\{f_0 f_1 \ldots f_{M-1}\}$	frame sequence (i.e. M symbol bits)

the output of the Gold Code Encoder 303 is a sequence with N x M terms having the following elements :

$$\{f_0 c_0, f_0 c_1, \ldots f_0 c_{N-1}, f_1 c_0 \ldots f_1\ c_{N-1} \ldots$$
$$\ldots f_{M-1} c_{N-1} \ldots f_{N-1} c_{M-1}\ \}$$

Hence, there are MN chips to modulate.

A modulator 304 modulates these MN chips output by the Gold Code Encoder 303 on to a carrier, which is output by clock 305. The modulator 304 generally used in CDMA systems such as IS95 is a linear QPSK modulator. However, according to the present invention, the modulator is a non-linear modulator such as that used in MSK modulation. In a preferred embodiment GMSK modulation is used The bandwidth of the signal output by the modulator 304 is directly related to the

spectrum of the pulses that are used to make up a lookup table 306. Conventionally, in CDMA, this lookup table would comprise data defining a root raised cosine. However, in this preferred embodiment of the present invention, the lookup table defines a different pulse whose shape may be one of the alternatives outlined below. The output of the modulator 304 is input to a digital to analogue converter 307. The analogue signal is then reconstructed by a reconstruction filter 308. A reconstruction filter might typically comprise a switch capacitor filter for performing some spectral shaping and an analogue filter, such as an RC filter network, for mainly dealing with residual spectral shaping. Once the signal has been reconstructed, it is input to a power amplifier 309, which amplifies the signal for transmission by the antenna 310.

The lookup table may define a Gaussian pulse shape, as is conventionally used in GMSK modulation. Alternatively, this shape may be approximated, by using one or more AM pulses according to Laurent's superposition theory, these pulses being a fixed family of pulses which are functions of cos and sin. However, in this preferred embodiment, the lookup table stores a new pulse shape which depends upon desired cost functions. The new pulse shape is determined by the following principles.

In prior art modulation schemes the pulse functions used to shape the data streams have had a predefined mathematical relationship.

For example:

root raised cosine

$$H(f) = 1 \qquad |f| < \propto$$

$$H(f) = \sqrt{\frac{1}{2}(1 - \cos(2\Pi(f - (T + \alpha))} \qquad \propto < |f| \leq T + \propto$$

$$= 0 \qquad |f| > T + \propto$$

for CDMA systems in which linear QPSK modulation is used and PDC and NADC systems in which OQPSK modulation is used.

Gaussian

$$H(f) = \frac{1}{\sqrt{2\Pi}\,\sigma}\, e^{-f^2/2a^2}$$

for GSM in which non-linear MSK modulation scheme is used.

With pulse shapes according to the conventional predefined mathematical relationships only one parameter is variable for a given energy level. For the gaussian pulse this is 'sigma' that varies the spread of the pulse allowing the bandwidth to alter at the expense of amplitude. For the root raised cosine the variable is 'alpha' that varies the frequency at which the cosine tail begins. This effects the bandwidth and consequently the power efficiency. The relationship between the cost parameters is well defined so as one improves the other declines in a determined fashion. There is no scope for improving both cost parameters.

For the mathematical modulating functions defined by a single parameter, the trade-offs achievable are those obtained by assigning a value to the parameter. The single variable of the mathematical function is set by the system designer to provide an acceptable balance in the defined relationship between the cost parameters.

In this aspect of the present invention, there is no predetermined mathematical relationship for the pulse shaper. The shape of the pulse is defined in order to meet desired cost parameters. There is freedom to select new pulse shapes that allow many cost parameters to be balanced against each other. The trade-off relationship between two parameters is no longer defined so restricted. This leads to a number of interesting possibilities. With the present invention it is not necessary for the pulse shape in MSK to be gaussian. Although this particular pulse shape optimizes performance in terms of power efficiency it is not optimal in terms of spectral efficiency. By deviating from a gaussian shape in the frequency domain the balance between BER, power efficiency and bandwidth alters.

Pulses in the MSK system, for example can be shaped to provide a desired balance between cost functions (e.g. BER, bandwidth, power efficiency, AFC) rather than being at the mercy of existing trade-offs when only the parameters of a given shape pulse such as a gaussian are modified. Cost functions are functions which are positive and get smaller the closer a system operates to the desired mode.

Simulations demonstrate that by experimenting with the pulse shape used for MSK modulation the spectral efficiency of the telecommunication system can be enhanced while retaining an acceptable power efficiency. This allows GMSK to be readily used in a CDMA system by modifying the pulse shape in the MSK modulation scheme to reduce the relevant cost function i.e. bandwidth or bit error rate.

Thus according to this embodiment, existing non-linear modulation schemes such as GMSK can use new pulse shapes to obtain performances that are better than those possible at present with conventional pulse shapes. By removing the strong links between particular modulation schemes and the current problems, (e.g. MSK and spectral inefficiency) the modulation schemes for particular systems can be selected on a different basis.

For example, by implementing CDMA using a MSK modulation scheme with a suitable pulse shape to meet the required cost function i.e. (CDMA criteria) a dual mode GSM CDMA terminal can be constructed using a single modulator (see figures 6 and 7). The pulse shape would be likely to differ for each system as the cost function (desired parameters) may also differ.

A method of adaptively shaping a pulse function depending upon cost functions, for storing in lookup table 306 is described below.

As mentioned above, only the Gaussian and root raised cosine pulses have been considered for use in modulators of telecommunications systems to date. Laurent has suggested that a Gaussian pulse can be approximated by the superposition of

AM pulses (C_0, C_1 ...etc.), these pulses being a fixed family of pulses which are functions of cos and sin. According to this aspect of the present invention, a totally different approach has been taken, as is outlined below.

Laurent's theory that a pulse can be approximated by the superposition of components has been implemented. However, instead of using this theory to approximate existing Gaussian pulses based on the fixed function components, Laurent's superposition expansion has been used as the basis of ascertaining a pulse shape which meets the criteria required by a particular communications system. This may be done as follows.

Firstly, the fixed function components in Laurent's superposition expansion are replaced by one or more functions representing respective unknown pulse components. Then cost functions are looked at (e.g. BER, bandwidth, amplitude, AFC). That is, the errors from the values that the particular system requires are considered. The weightings of the cost functions can be varied so as to tailor the results. Values for each function are then determined, for example using an optimiser, which minimise these cost functions and thus give a pulse shape which meets the specified system requirements.

Preferably two functions are used as this provides more optimal pulse shaping than just using one function.

More specifically, the method can be implemented as follows :

Firstly, Laurent's formula is considered. According to Laurent's formulation:

$$S_{NT+\Delta T}=\sum_{K=0}^{M-1}\sum_{n'=0}^{L_{k-1}} J^{A}{}_{K,N-n'}C_{K,n'T+\Delta T} \quad \text{- Equation (1)}$$

where S(t) is the signal at time t $A_{K,N}=\sum_{n=-\infty}^{N} a_n-\sum_{i=1}^{L-1} a_{N-i}\cdot\alpha_{K,i}$

$$A_{K,N} = A_{O,N} - \sum_{i=1}^{L-1} a_{N-i} \cdot \propto_{K,i.}$$

$$C_K(t) = S_0(t) \times \prod_{i=1}^{L-1} S_{i-L \cdot \propto_{K,i}}(t) \qquad (0 \le K \le M-1)$$

Instead of using Laurent's pulses, $C_{K,n'}$, we wish to use an alternative pulse, PULSE $K_{K,n}$, which is as yet unknown, but for which we wish to determine an appropriate value depending upon requisite error function requirements.

Substituting this in equation 1 gives :

$$S_{NT+\Delta T} = \sum_{K=0}^{M-1} \sum_{n'=0}^{L_{k-1}} J^{A_{K,N-n'}} PULSE_{K,n'T+\Delta T} \quad \text{- Equation (2)}$$

where J= $\sqrt{-1}$

As mentioned above, PULSE is unknown as yet, but is, in this embodiment it is read, non zero and of maximum length 8.

In this embodiment we choose to use two components (PULSE [0] and PULSE [1]) to build up S. Hence M = 2. Expanding equation (2) for M=2 and replacing the function A_K with a function of the bit streams $\propto_1, \propto_2 \cdots$, gives :

$$J^{A_{O,N-5}} \Big(J^{(\propto_{N-4}+\propto_{N-3}+\propto_{N-2}+\propto_{N-1}+\propto_N)} \text{ Pulse } [0][\delta T] +$$

$$J^{(\propto_{N-4}+\propto_{N-3}+\propto_{N-2}+\propto_{N-1})} \text{ Pulse } [0][T+\delta T] +$$

$$J^{(\propto_{N-4}+\propto_{N-3}+\propto_{N-2})} \text{ Pulse } [0][2T+\delta T] +$$

$$J^{(\propto_{N-4}+\propto_{N-3})} \text{ Pulse } [0][3T+\delta T] + J^{\propto_{N-4}} \text{ Pulse } [0][4T+\delta T] +$$

Pulse [0] $[5T+\delta T]$ + $J^{(-\propto N-5)}$ Pulse [0] $[6T+\delta T]$ +

$J^{(-\propto N-5-\propto N-6)}$ Pulse [0] $[7T+\delta T]$ +

$J^{(-\propto N-5-\propto N-6-\propto N-7)}$ Pulse [0] $[8T+\delta T]$ +

$J^{(\propto N-4+\propto N-3+\propto N-2+\propto N-1+\propto N \propto -a N-1}$ Pulse [1] $[\delta T]$ +

$J^{(\propto N-4+\propto N-3+\propto N-2+\propto N-1-\propto N-2)}$ Pulse [1] $[T+\delta T]$ +

$J^{(\propto N-4+\propto N-3+\propto N-2+\propto N-3)}$ Pulse [1] $[2T+\delta T]$ +

$J^{(\propto N-4+\propto N-3-\propto N-4)}$ Pulse [1] $[3T+\delta T]$ +

$J^{(\propto N-4-\propto N-5)}$ Pulse [1] $[4T+\delta T]$ + $J^{(-\propto N-6)}$ Pulse [1] $[5T+\delta T]$ +

$J^{(-\propto N-5+\propto N-7)}$ Pulse [1] $[6T+\delta T]$)

-Equation (3)

Since $\propto$ denotes a bit, it must be plus or minus 1. Hence each term in equation (3) can be identified as to whether it is real or imaginary (assuming that the pulse function is real).

eg: Taking the first term of the equation :

$J\ (\propto_{N-4}+\propto_{N-3}+\propto_{N-2}+\propto_{N-1}+\propto_{N})$

$\propto_{N-4}, \propto_{N-2}, \propto_{N}$ = odd → imaginary

$\propto_{N-3}, \propto_{N-1}$ = even → real.

Hence it is possible to calculate the absolute value of this expression as a function of the bits $(\propto s)$. A decision to be made is what $\propto$ is sent at time N. (In an ideal system this will be the signal received at baseband).

Looking at equation 3 (e.g. for a simple receiver), it can be deduced that the bit $\propto_{N-4}$ is transmitted at time $(N+4)T$ as it is on its own. It is imaginary, and the

interfering (ie other imaginary) pulses must be taken into account. The real terms in this expression can be totally ignored both for the interfering terms and the absolute value of the pulses.

The interference should be minimised. The BER performance can, for example, be improved by making the terms Pulse[0]at $(N+4)T$ large compared to the absolute value of all the other terms.

Therefore, given an $\propto$ sequence of :

$$\{\propto_N, \propto_{N-1}, \cdots \quad \propto_{N-7}\} = \{1,1,1,1,1,1,1,1\},$$

the absolute value of the pulse at time ΔT can be calculated in terms of the unknown pulses. The absolute value of the interfering terms at time ΔT can also be calculated in terms of the unknown pulses. This is performed for every possible combination of 1,-1 for $\propto_N$ to $\propto_{N-7}$ (ie all 2^8 = 256 possibilities). For each possibility an expression both for interfering terms and absolute value are obtained.

In this embodiment, the pulse is required to meet certain criteria with regard to power, BER, AFC and bandwidth. Hence, error functions for these are determined.

Given an oversampling of 8, ΔT can take on the following values :

$$\Delta T = \{0, \frac{T}{8}, \frac{2T}{8}, \frac{3T}{8} \ldots \frac{7T}{8}\}$$

Clearly, the oversampling rate can be altered depending upon the level of pulse sampling required.

The amplitude and BER costs are calculated for ΔT taking each of the above values. The total cost for each is the addition of all the 8 expressions obtained over the possible sequences.

Cost (Error) Functions

(i) Amplitude Error Function

Given a constant amplitude of 1, the error in amplitude can be given by :

$$\{ \text{absolute value}\ ^{2} - 1^{2} \}^{2}$$

(ii) BER Error Function

To calculate this, the amount of noise needs to be determined. This is given by :

$$\{ \text{absolute value of interfering regions} \}^{2}$$

(iii) Energy Error Function

Required energy - sum of the square of the sample points.

(iv) Bandwidth Error Function

In order to estimate the bandwidth of the pulses, the derivative of the pulse functions (which at this stage are still unknown) are required. This derivative can be approximated as being the difference between two adjacent pulse values. The bandwidth for a pulse, is given by :

$$\text{sum} \quad \{ \text{derivative at the sample points} \}^{2}$$

This can be determined as follows :

A pulse width of 8T has been assumed and we have oversampled the pulse by 8. According to Laurent :

Pulse $[0][t]$ is non zero for $0 \le t \le 9T$

Pulse $[1][t]$ is non zero for $0 \le t \le 7T$

The unknown pulses are :

Pulse[0][$mT+\frac{nT}{8}$] for m = 0,1,2...8

n = 0,1,2...7

Pulse[1][$mT+\frac{nT}{8}$] for m = 0,1,2...7

n = 0,1,2...7

For convenience denote :

Pulse[0][$mT+\frac{nT}{8}$] by $x_{0,m\cdot 8+n}$

Pulse[0][$mT+\frac{nT}{8}$] by $x_{1m\cdot 8+n}$

Then for example :

Pulse[0][$2T+\frac{3T}{8}$] = $x_{0,19}$

Pulse[1][$2T+\frac{3T}{8}$] = $x_{1,19}$

Adjacent sample points have adjacent numbers and the set of unknowns becomes :

$x_{0,i}$ where $i = 0,1,2...71$

and

$x_{1,i}$ where $i = 0,1,2...55$

Consequently, the approximate bandwidths for pulse [0] is as follows :

$$\text{Pulse [0] : } Sum\frac{(x_{0,i+1}-x_{0,i})^2}{T/8} \quad \text{for } i = 0 \text{ to } 70 \qquad \text{-(a)}$$

In the present embodiment, the bandwidth for the second component is to be determined, then a similar expression needs to be determined for PULSE [1]. This is as follows :

Pulse [1] : $Sum\frac{(x_{1,i+1} - x_{0,i})^2}{T/8}$ for i = 0 to 54

-(b)

Similarly, expressions can be determined where further pulses are required to further improve the integrity of the signal which is being sent.

Total bandwidth for the pulse composed of the two components = (a) + (b)

The pulse can be specifically designed based on system requirements by weighting the above error functions (for example 0.3 for power, 0.3 for BER and 0.4 for bandwidth or if a system requires only, for example, bandwidth considerations, 0 for power and BER and 1 for bandwidth). More weight can be added to whatever is causing a problem. The only restriction is that the total weighting must equal + 1.

Now the total error function is expressed in terms of the unknowns, namely, $x_{0,i}(i = 0$ to 71) and $x_{i,j}(i = 0$ to 55). To determine appropriate values for the unknowns, and thus deduce the pulse shapes, this expression is minimised using a conventional off-the-shelf optimiser, for example.

Figure 4 is a block diagram of a spread spectrum receiver with a non-linear demodulator. In this embodiment, the receiver complements the CDMA transmitter of figure 3. It comprises an antenna for receiving a spread signal, frequency downconverting circuitry 401, analogue to digital converter 402, means for storing the receiver's code and a despreader 404. The despreader comprises means 405 for transforming the receiver's code according to the present invention, a correlator 406 for correlating the received signal and the transformed code, and a comparator 407 for determining the sign of the received signal. Operation of the receiver may be as described with reference to figures 2, 8 and 9 above.

The code transformer 405 may solely comprise the transformation for detecting a + 1 bit. In this event, the comparator assumes it is a +1 if the value of the correlated signal output by correlator 406 is above a certain threshold, and –1 if it is below this

threshold. However, this type of receiver requires more complex demodulation. Simpler demodulation is possible if the transformer 405 also has a transformation for detecting a –1. In this event the comparator determines the sign which produces the largest value. That value should be well above the noise floor, and hence, no complex demodulation is required to determine whether in fact the received signal is a –1, intended for that receiver, or noise resulting from signals for other receivers.

Figures 5a and 5b show a CDMA receiver according to a preferred embodiment, which complements a transmitter which transmits a signal constructed using the superposition of a plurality of amplitude modulated pulses.

As can be seen from figure 5(a), the frequency downconverting circuitry 401 comprises at least 1 IF stage 501, mixers 502a, 502b and low pass filters 503a and 503b. A received signal is put through the IF stage(s) 501 to reduce its frequency to a base band frequency and then the signal is split into its I and Q comonents and the carrier is removed from the signal, using mixers 502a and 502b and low pass filters 503a and 503b. The signal is then converted from an analogue signal into a digital signal by A/D converters 504a and 504b and forwarded to the demodulator stage 404. Figure 5(b) shows this demodulator stage in more detail.

The code transformer 405 transforms the Gold Code to detect a +1 and a –1 for both of the amplitude modulated pulses which make up the received signal. Exemplary transformations are given below:

Transformation 1 (TI) 505a (to detect +1 symbol, for 1st AM pulse).

$y_i = (-1)^i$ for $i = 0, 1, 2, \ldots N - 1$

Given code $(C_0, C_1, \ldots C_{N-1})$ gold code where N is the number of elements in the sequence.

$a_i = 1$ if $C_i = 1$; for $i = 0, \ldots N - 1$

$a_i = -1$ if $C_i = 0$; for $i = 0, \ldots N - 1$

$b_0 = a_0$;

$b_i = b_{i-1} + a_i$ for $i = 1, 2, \ldots N - 1$;

$d_i = y_i\, i^{b_i}$ for i = 0, 1, 2,... N – 1 and

$i = \sqrt{(-1)}$

There is another transformation that can also be used

Transformation 1b (to detect + 1 symbol, for 1st AM pulse) using the same notation for d_i

$d_i = i^{-b_i}$ for i = 0, 1, 2, ... N – 1 and

Transformation 2 (T2) 505b (to detect –1 symbol for 1st AM pulse)

$y_i = (-1)^i$ for i = 0, 1, 2, ... N –1

Given code $\{C_0, C_1, \ldots C_{N-1}\}$ where N is the number of elements in the sequence

$a_i = -1$ if $C_i = 1$ for i = 0, ... N –1

$a_i = 1$ if $C_i = 0$ for i = 0, ... N –1

$b_0 = a_0$;

$b_i = b_{i-1} + a_i$ for i = 1, 2, ... N – 1;

$d_i = y_i\, i^{b_i}$ for i = 0, 1, 2,... N – 1 and

$i = \sqrt{(-1)}$

There is another transformation than can also be used

Transformation 2 b (to detect – 1 symbol, for 1st AM pulse)

$d_i = i^{-b_i}$ for i = 0, 1, 2, ... N – 1

Transformation 3 (T3) 505c (to detect +1 symbol for 2nd AM pulse)

$y_i = (-1)^i$ for i = 0, 1, 2, ... N –1

Given code $\{C_0, C_1, \ldots C_{N-1}\}$ where N is the number of elements in the sequence

$a_i = 1$ if $C_i = 1$ for i = 0, ... N –1

$a_i = -1$ if $C_i = 0$ for i = 0, ... N –1

$b_0 = a_0 -/+ a_N$

$b_i = b_{i-1} + a_i - a_{i-1}$ for i = 1, 2, ... N – 1;

$d_i = y_i\, i^{b_i}$ for i = 0, 1, 2,... N – 1 and

$i = \sqrt{(-1)}$

There is another transformation than can also be used

Transformation 3 b (to detect – 1 symbol, for 2nd AM pulse)

$d_i = \mathrm{i}^{-b_i}$ for i = 0, 1, 2, ... N – 1

Transformation 4 (to detect +1 symbol for 2nd AM pulse)

$y_i = (-1)^i$ for i = 0, 1, 2, ... N –1

Given code $\{C_0, C_1, \ldots C_{N-1}\}$ where N is the number of elements in the sequence

$a_i = -1$ if $C_i = 1$ for i = 0, ... N –1

$a_i = 1$ if $C_i = 0$ for i = 0, ... N –1

$b_0 = a_0 -/+ a_N$

$b_i = b_{i-1} + \quad a_i - a_{i-1}$ for i = 1, 2, ... N – 1;

$d_i = y_i\, \mathrm{i}^{b_i}$ for i = 0, 1, 2,... N – 1 and

$\mathrm{i} = \sqrt{(-1)}$

There is another transformation than can also be used

Transformation 4 b (to detect – 1 symbol, for 2nd AM pulse)

$d_i = \mathrm{i}^{-b_i}$ for i = 0, 1, 2, ... N – 1

Likewise, the correlator 406 performs a correlation of each transformed code with the respective pulse of the received signal. For example, the transformed Gold Code associated with the first AM pulse for detecting a + 1 is correlated with the 1st AM pulse of the received signal (x_{i1},x_{q1}) by correlator 506a. The absolute value z_1 of the correlated signal y_1 is forwarded to the comparator 407. The same stages occur for the first AM pulse for detecting a – 1, and for the second AM pulse for detecting a + 1 and for detecting – 1.

The comparator 407 determines whether the received signal is +/- 1. This is achieved by a comparison of the absolute values $(z_l - z_4)$ received from the comparator with expected absolute of the values $(E_1 - E_4)$ assuming the received signal is of the sign being detected. The values $E_1 - E_4$ can be precalculated and stored in the receiver. In this embodiment, if the received signal is a + 1, then the value of z_1 and z_3 will be close to their associated expected values E_1 and E_3, so that the values of h_1 and h_3 will be small. In contrast z_2 and z_4 will be much smaller in value than E_2 and E_4, so that the values of h_2 and h_4 will be large. Hence, the comparator determines that a + 1 is received as $h_1 + h_3 < h_2 + h_4$. Alternatively, if a –

is received, the value of z_2 and z_4 will be close to their expected values E_2 and E_4, so that the values of $h_2 + h_4$ will be small, whereas z_1 and z_3 will be much smaller than E_1 and E_3, so that h_1 and h_3 will be larger values. Hence the comparator determines that a -1 is received as $h_2 + h_4 < h_1 + h_3$. When there is little interference on the chanel, the receiver need only perform the correlations for the first pulse.

Similar transformations can be provided for further AM pulses, should they be so desired. However, use of the first two pulses is generally acceptable as most energy is found in these pulses.

Figure 6 shows a dual mode GSM/CDMA transmitter. This transmitter is provided with a common modulator 604. This is possible because the present invention provides compatibility between a non linear modulation scheme and a spread spectrum scheme (in this case GMSK and CDMA). This embodiment provides a preferred solution as the cost function restraints of a particular modulation scheme have been reduced by the transmitter having two lookup tables 606a and 606b, which, in this embodiment respectively define pulse shapes which meet the cost function requirements of GSM and CDMA. As can be seen a number of components can be used for both GSM and CDMA operations and where two components are required a switch is included, the switch between them depending on the operation of the transmitter. For example, if in CDMA mode the bit sequence 601 would need to be encoded by a gold code encoder 603. Hence the switch would make a connection with this gold code encoder, whereas if in GSM mode it would switch straight through to the modulator. Similarly if in GSM mode, the pulse shaping is provided by GSM lookup table 606a and switch 611 provides a connection so a bit sequence can be shaped according to the data in this lookup table. Finally switch 612 is provided so that the power amplifier is connected to the filter network for the appropriate mode of operation of the transmitter.

Figure 7 shows a dual mode GSM/CDMA receiver to complement the transmitter of Figure 6.

A complex number sequence 701 received by the receiver is switched between a GSM demodulator 703a and a CDMA demodulator 703b, depending on the mode of operation of the receiver. GSM demodulator 703a is a conventional GSM demodulator, comprising a Viterbi decoder, for example. The CDMA demodulator 703b, on the other hand, is a demodulator according to the present invention. That is, one which demodulates a signal spread by a code and modulated according to a non-linear modulation scheme, such as that shown in Figure 4 and/or 5. In this way, the original bit sequence 704 transmitted is determined by the receiver.

Preferably the dual mode transmitter and receiver of Figures 6 and 7 manipulate a signal constructed using the superposition of two (or more) pulses. Whilst this involves more processing by the receiver, there is a better integrity of what is being sent. This is especially useful for narrow band GSM (e.g. BT product of O.I5), in which the first pulse does not provide a sufficient approximation, as the second pulse contains a significant amount of information.

Figures 10a to 10d show a +1 bit having been despread by a receiver arranged to detect a +1 bit at different synchronisation instants.

Figure 10a shows the real value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of +1 by a gold code sequence and transmitting the resulting bits employing GMSK type modulation using two pulses), with the same code rotated by x bits and then transformed to receive a bit with a value of +1, using the first pulse as x varies between 0 and 250.

Figure 10b shows the real value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of + 1 by a gold code sequence and transmitting the resulting bits empoying GMSK type modulation using two pulses), with the same code rotated by x bits and then transformed to receive a bit with a value of +1, using the first pulse as x varies between 0 and 20.

Figure 10c shows the imaginary value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of +1 by a gold code sequence and transmitting the resulting bits employing GMSK type modulation using two pulses), with the same code rotated by x bits and then transformed to receive a bit with a value of +1, using the first pulse as x varies between 0 and 20.

Figure 10d shows the absolute value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of +1 by a gold code sequence and transmitting the resulting bits employing GMSK type modulation using two pulses), with the same code rotated by x bits and then transformed to receive a bit with a value of +1, using the first pulse as x varies between 0 and 20.

The receiver may use, for example, the transformation shown in figure 8a for transforming the code used to despread the data signal (+1 bit). The transformation may be implemented using the transformation T1 or T1b given above in respect of the transformation 505A. As can be seen from figures 10a to d, a +1 bit was detected and an information bit 101 was successfully identifiable, as its amplitude is much greater than a noise floor 102.

In order for a signal to be successfully despread, the code used by the despreader or decoder must correspond to that used by the spreader or encoder. Figure 11 shows the autocorrelation of a signal having been spread by one code, and then despread by a receiver using a transformation of another code. More specifically, Figure 11 shows the absolute value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of +1 by a gold code sequence and transmitting the resulting bits employing GMSK type modulation using two pulses), with a different code rotated by x bits and then transformed to receive a bit with a value of +1, using the first pulse as x varies between 0 and 250. As can be seen from figure 11, the signal in this case is not distinguishable from the noise.

Figure 12a illustrates a -1 bit despread by a receiver arranged to detect a -1 bit at different synchronisation instants. That is, it shows the absolute value of the

correlation of the baseband signal (which was obtained by spreading a bit with value of –1 by a gold code sequence and transmitting the resulting bits employing GMSK type modulation using two pulses), with a different code rotated by x bits and then transformed to receive a bit with a value of –1, using the first pulse as x varies between 0 and 250.

The receiver may use, for example, a transformation such as that shown in figure 8b for transforming the code to despread the data signal (-1 bit). This may be implemented using transformations T2 or T2b given above in respect of the transformation 505B. As can be seen, a -1 bit was determined, which is much greater in amplitude than the corresponding noise floor.

Figure 12b, on the other hand, shows the output of a receiver arranged for detecting a +1 bit (for example that used with the receiver output shown in figure 10) when trying to despread a -1 bit. That is, it shows the absolute value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of –1 by a gold code sequence and transmitting the resulting bits employing GMSK type modulation using two pulses), with a different code rotated by x bits and then transformed to receive a bit with a value of +1, using the first pulse as x varies between 0 and 250. As can be seen, in this embodiment, the bit is not decipherable from the noise.

Figure 13 illustrates the distinction between the two transformation methods described above with respect to figure 5. Figures 13a and b show the real value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of +1 by a training sequence found in a GMSK frame and transmitting the resulting bits employing GMSK type modulation using two pulses), with the same training sequence rotated by x bits and then transformed to receive a bit with a value of +1, using the first pulse as x varies between 0 and 25, for different transformation methods, T1 and T1b respectively.

Figure 13c and figure 13d show the real value of the correlation of the baseband signal (which was obtained by spreading a bit with a value of +1 by a training sequence found in a GMSK frame and transmitting the resulting bits employing GMSK type modulation using two pulses), with the same training sequence rotated by x bits and then transformed to receive a bit with a value of +1 using the second pulse as x varies between 0 and 25, for different transformation methods, T3 and T3b respectively. As can be seen from figures 13a and 13b, when the data signal is approximated by a first Laurent type pulse, both transformations T1 and T1b provide comparable results. However, in narrow band systems in particular, the second pulse of the Laurent approximation contains a significant amount of information which needs to be extracted to give an acceptable integrity. As can be seen from figures 13c and d, in this case, it is preferable that the transformation T3 is used as opposed to T3b.

Annex 1 provides further background with regard to figures 10 to 13, and also mathematical simulations of the invention.

The present invention includes any novel feature or combination of features disclosed herein either explicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed.

In view of the foregoing description it will be evident to a person skilled in the art that various modifications may be made within the scope of the invention. For example, it is evident that references to a dual mode phone include a phone having two or more modes.

Also, it is evident that the transformations described may be modified as follows within the scope of the invention. Firstly, the sequences resulting from the transformations may be cyclically rotated, and secondly the elements of the sequence obtained from the transformations may be multiplied by a constant (real or complex).

Claims

1. A method of despreading a data signal which has been spread by a code and modulated according to a non-linear modulation scheme, the method comprising:

transforming the code into a succession of phasor positions; and

correlating the received spread signal with the transformed code.

2. A method as claimed in claim 1 for despreading a data signal constructed using the superposition of N amplitude modulated pulses, wherein the steps of transforming and correlating are performed for M pulses.

3. A method of demodulating a data signal which has been spread by a code and modulated according to a non-linear modulation scheme, comprising:

despreading the received signal according to the method of claim 1 or 2, assuming the data signal is positive;

despreading the received signal according to the method of claim 1 or 2, assuming the data signal is negative; and

comparing the correlated signals to determine the sign of the received data signal.

4. A method of transmitting and receiving a data signal in a spread spectrum telecommunications system, the method comprising:

spreading a data signal to be transmitted by a code;

modulating the spread signal using a non-linear modulation scheme;

transmitting the modulated signal;

receiving the modulated signal;

transforming the code into a succession of phasor positions; and

correlating the received signal with the transformed code.

5. A method as claimed in claim 4, comprising:

performing the transforming and correlating steps assuming the data signal is positive;

performing the transforming and correlating steps assuming the data signal is negative; and

comparing the correlated signals to determine the sign of the received data signal.

6. A method as claimed in claim 4 or 5, further comprising shaping the signal to be transmitted.

7. A method as claimed in claim 6, wherein the step of shaping the signal comprises constructing the signal using the superposition of N amplitude modulated pulses, and the steps of transforming and correlating are performed for M pulses.

8. A method as claimed in claim 2, 3 when dependent upon 2 or 6, wherein N=M=2.

9. A method as claimed in claim 2, 3 when dependent upon 2 or 6, wherein N>M.

10. A method as claimed in claim 9, wherein N=4 and M=2.

11. A method as claimed in claim 2, 3 when dependent on 2, 7, 8, 9 or 10 wherein the pulses have a pulse function, the relationship between frequency and amplitude of which have been determined by:

defining desired cost parameters; and

defining the amplitude of the pulse function over a range of frequencies in dependence on the desired cost parameters.

12. A despreader for despreading a data signal which has been spread by a code and modulated according to a non-linear modulation scheme, the despreader comprising:

means for providing the code transformed into a succession of phasor positions; and

a correlator for correlating the spread signal with the transformed code.

13. A despreader as claimed in claim 12, for despreading a data signal constructed using the superposition of N pulses, comprising transformed code providing means and a correlator for M pulses.

14. A demodulator for demodulating a data signal which has been spread by a code and modulated according to a non-linear modulation scheme, the receiver comprising:

a first despreader as claimed in claim 9 or 10, for despreading the received signal assuming the data signal is positive;

a second despreader as claimed in claim 9 or 10, for despreading the received signal assuming it is negative; and

a comparator for comparing the correlated signals output by the first and second despreaders to determine the sign of the received signal.

15. A receiver for a communication device comprising a despreader as claimed in claim 9 or 10 or a demodulator as claimed in claim 11.

16. A transceiver for a communication device comprising:

a transmitter having a spreader for spreading a data signal by a code, a non-linear modulator for modulating the spread signal and means for transmitting the modulated spread signal; and

a receiver as claimed in claim 15.

17. A communication device operable in a communication system, comprising a transceiver as claimed in claim 16.

18. A device as claimed in claim 16 or 17, wherein the transmitter comprises a signal shaper for shaping the signal to be transmitted.

19. A device as claimed in claim 18, wherein the signal shaper constructs the signal using the superposition of N amplitude modulated pulses, and the despreader comprises code transforming means and a correlator for M pulses.

20. A device as claimed in claim 13, any of claims 14 to 18 when dependent upon claim 13, or claim 19, wherein N=M=2.

21. A device as claimed in claim 13, any of claims 14 to 18 when dependent upon claim 13, or claim 19, wherein N>M.

22. A device as claimed in claim 21, wherein N=4 and M=2.

23. A device as claimed in any of claims 19 to 22, wherein the pulses have a pulse function, the relationship between frequency and amplitude of which have been determined by:

defining desired cost parameters; and

defining the amplitude of the pulse function over a range of frequencies in dependence on the desired cost parameters.

24. A dual mode receiver operable in a first mode in a spread spectrum telecommunications system and a second mode in a telecommunications system which uses a non-linear modulation scheme, the receiver comprising:

a first demodulator for demodulating a received signal in accordance with the non-linear modulation scheme in the first mode of operation, comprising a despreader having means for providing the code transformed into a succession of phasor positions, and a correlator for correlating the received signal with the transformed code; and

second receiver means comprising a demodulator for demodulating a received signal in accordance with the non-linear modulation scheme in the second mode of operation.

25. A device as claimed in claim 24, wherein the first demodulator comprises:

a first despreader for despreading the received signal assuming the data signal is positive;

a second despreader for despreading the received signal assuming the data signal is negative; and

a comparator for comparing the correlated signals output by the first and second despreaders to determine the sign of the received data signal.

26. A dual mode transmitter operable in a first mode in a spread spectrum telecommunications system and a second mode in a telecommunications system which uses a non-linear modulation scheme, the device comprising a modulator for modulating a data signal with a carrier signal in accordance with the non-linear modulation scheme in both the first and second modes of operation and means for spreading the data signal prior to modulation in the first mode.

27. A dual mode communication device, comprising a receiver as claimed in claim 24 or 25 and a transmitter as claimed in claim 26.

28. A device as claimed in claim 26 or 27, comprising a signal shaper for shaping the signal to be transmitted.

29. A device as claimed in claim 28, wherein the signal shaper constructs the signal using the superposition of N amplitude modulated pulses, and the despreader comprises code transforming means and a correlator for M pulses.

30. A device as claimed in claim 29, wherein N=M=2.

31. A device as claimed in claim 29, wherein N>M.

32. A device as claimed in claim 31, wherein N=4 and M=2.

33. A device as claimed in any of claims 29 to 32, wherein the pulses have a pulse function, the relationship between frequency and amplitude of which have been determined by:

defining desired cost parameters; and

defining the amplitude of the pulse function over a range of frequencies in dependence on the desired cost parameters.

34. A device as claimed in any of claims 24, 25, 27 to 33, operable in the first mode in a CDMA system.

35. A device as claimed in any of claims 24, 25, 27 to 34, operable in the second mode in a system which uses phase modulation.

36. A device as claimed in any of claims 24, 25, 27 to 35, operable in the second mode in a TDMA system.

37. A method of despreading a signal substantially as hereinbefore described with reference to Figure 8 and/or 9 of the accompanying drawings.

38. A despreader and/or demodulator substantially as hereinbefore described with reference to any one, or any combination, of Figures 2, 4, 5(a) or 5(b), with or without reference to any one, or any combination, of Figures 3 and 6 to 13 of the accompanying drawings.

39. A receiver comprising a despreader and/or demodulator as claimed in claim 38.

40. A transceiver comprising a receiver as claimed in claim 39.

41. A dual mode transmitter substantially as hereinbefore described, with reference to Figure 6 of the accompanying drawings.

42. A dual mode receiver substantially as hereinbefore described with reference to Figure 7 of the accompanying drawings, either with or without reference to any one, or any combination, of Figures 4, 5(a) and 5(b).

Abstract

A radio telephone

A method is provided for despreading a signal which has been spread by a code and modulated according to a non-linear modulation scheme. The method comprises transforming the code into a succession of phasor positions (405) and correlating the spread signal with the transformed code (406). A despreader, demodulator and communications device implementing such a method are also provided.

[Fig. 4]

ANNEX 1.

```
Needs["RingFunctionsDiffEncoded`"]

Names["RingFunctionsDiffEncoded`*"]

{AllGoldSequences, AutocorrelationSequence, AutomorphismSigma, CodingTransform,
 CodingTransformNew, CrosscorrelationSequence, CyclicMultiplativeGroup,
 DropLeadingZeros, GoldSequence, GraeffeMethod, InitialConditions, MinimumPoly,
 ModuloMultplication, PolynomialMultiplication, PossibleDivisors, RingDivision,
 RingPower, SequenceGenerator, SpecifiedGoldSequences, TraceRepresentation,
 TupleRepresentation, TwoAdicExpansion, UnitsRing, ZeroPad, ZeroSequences, τ, c}

Needs["LaurentFunctions`"]

RuleDelayed::rhs : Pattern t_ appears on the right-hand side of rule
   PhaseAngle[L_][t_] :> (PhaseAngle[L][t_] = Module[{x1, x2, x3, x4, x5, x6}, <<1>>]).

Needs["LaurentNotationTest`"]

Needs::nocont : Context LaurentNotationTest` was not created when Needs was evaluated.
```

Information on the functions used can be obtained using help.

```
Names["LaurentFunctions`*"]

{AKN, AlphaKI, AKNInitialStateSetUp, BT, FiltPulse, h, hFiltered, InitialState, J,
 LaurentC, LaurentLK, LaurentS, M, ModulatingPulse, ModulationIndex, Modulator,
 NumberOfCurves, PhaseAngle, PhaseAngleFast, Receiver, ReceiverProper, S,
 SamplingInterval, StartingQuadrant, SyncSample, T, C, Φ, ψ}
```

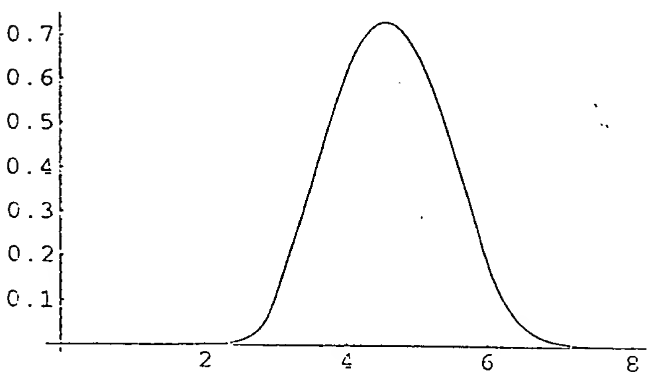
$T := \frac{3}{812500}$ GSM value of T

BT := 0.3

$\text{ModulationIndex} := \frac{1}{2}$

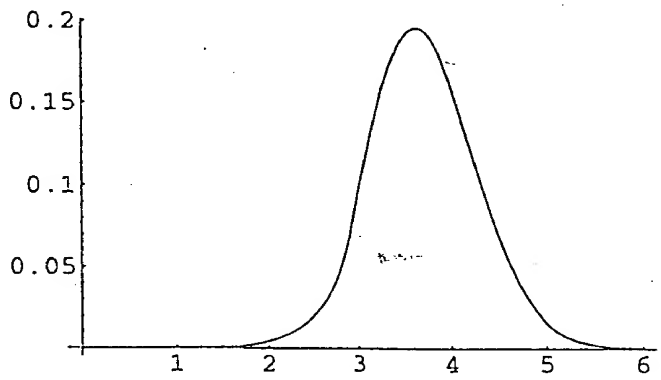
Example of optimal pulse shape.

```
<< ModulatorData.m;

<< OptimalPulseShapes.m;

Plot[OptPulse[L][0][t], {t, 0 , 8}]
```




```
- Graphics -

Table[
```

```
Plot[OptPulse[L][1][t], {t, 0 , 6}]
```




```
- Graphics -
```

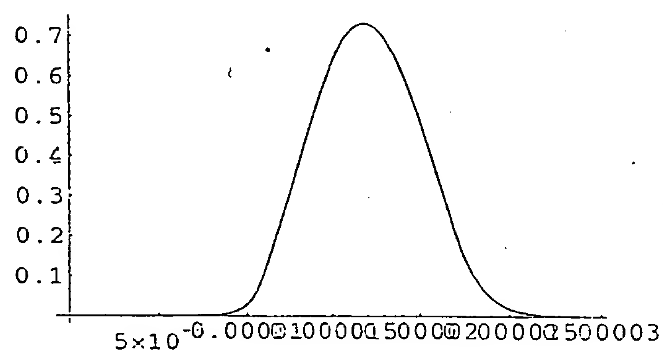
```
T
```

$\frac{3}{812500}$

The unit of time is T = 1 for OptPulse. We scale the Pulses to $T = \frac{3}{812500}$ for the unit of time.

```
OptPulseScaled[8][0][t_] := OptPulse[L][0][t / T]
OptPulseScaled[8][1][t_] := OptPulse[L][1][t / T]
```

```
Plot[OptPulseScaled[L][0][t], {t, 0 , 8 T}]
```

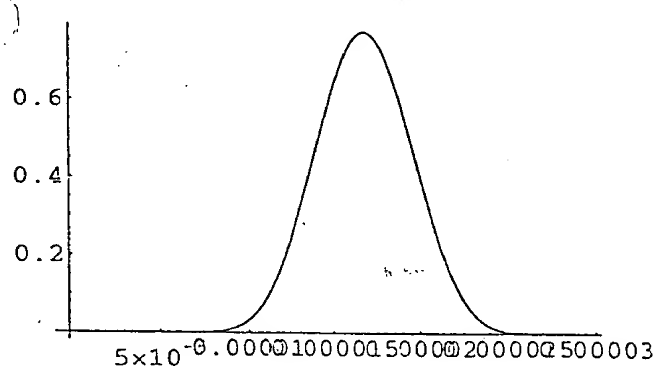



```
- Graphics -
```

```
RandomBitSeq
```

```
{1, 1, -1, -1, -1, 1, 1, -1, 1, -1, 1, -1, 1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, 1, 1, 1,
 -1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1, -1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1, 1, -1,
 1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1, -1, -1, -1, 1, -1,
 1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1}
```

```
Plot[FiltPulse[L][0][t], {t, 0, 8 T}]
```




```
- Graphics -
```

The Gold Sequence Set

We generate the sequences using the method specified by Serdar Boztas and P Vijay Kumar in Ref [1]. The numbering of the sequences is the one used in the paper. We generate a small subset of the sequences. There are $2^{10}+1$ sequences with the quaternary polynomial used. Given any binary primitive polynomial, we can generate the corresponding quaternary polynomial.

```
Goldseqlist =
 SpecifiedGoldSequences[{1, 3, 2, 1, 0, 3, 0, 0, 1}][{1, 2, 3, Last - 1, Last}];
```

The last sequence of seqlist has nice autocorelation properties.

The last sequence is in fact a m–sequence of length 1023 bits.

```
Goldseqlist // Last

{1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0,
 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0,
 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0,
 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1,
 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0,
 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0}

AutocorrelationSequence[Goldseqlist // Last]

{255, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}
```

The third sequence in the list has the following autocorrelation

```
AutocorrelationSequence[Goldseqlist // #[[3]]&]
```

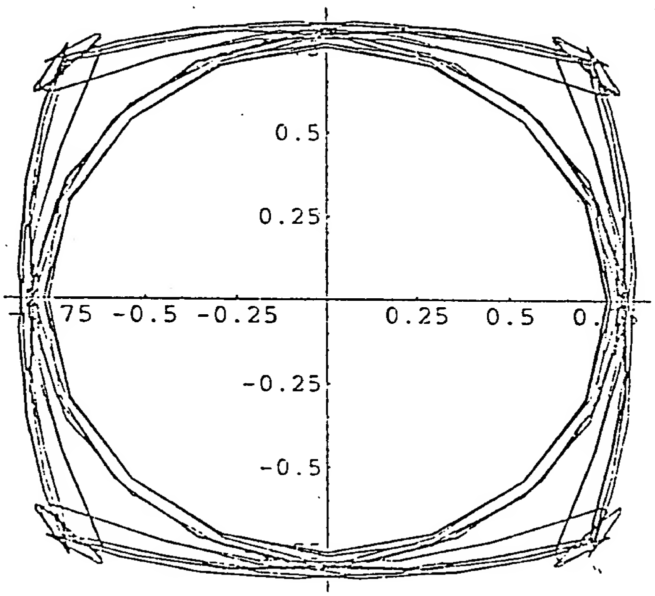
```
{255, -1, -1, -1, -1, -1, -1, -17, -1, -1, -1, -17, -1, -17, -17, -17, -1, -17, -1, -1,
 -1, -1, 15, -1, -1, 15, 15, -17, -17, -1, -17, 15, -1, -1, -17, 15, -1, 15, -1, 15, -1,
 15, -1, -1, 15, -17, -1, 15, -1, -1, -17, -17, -17, -1, -17, 15, 15, 15, -1, -1, 15,
 -17, 15, -1, -1, -1, -1, 15, 15, -1, 15, -1, -1, -17, -17, 15, -1, -1, 15, 15, -1, -1,
 -17, -1, -1, -17, -1, -1, 15, -1, 15, -17, -1, -1, -17, -1, -1, 15, -1, -17, 15, -1, 15,
 -1, 15, 15, -1, 15, -17, 15, 15, -1, 15, -1, 15, -17, -1, -1, -1, 15, -17, -17, 15, -1,
 -17, -1, -1, -1, -1, -1, -1, -17, -1, 15, -17, -17, 15, -1, -1, -1, -17, 15, -1, 15, -1,
 15, 15, -17, 15, -1, 15, 15, -1, 15, -1, 15, -17, -1, 15, -1, -1, -17, -1, -1, -17,
 15, -1, 15, -1, -1, -17, -1, -1, -17, -1, -1, 15, 15, -1, -1, 15, -17, -17,
 -1, -1, 15, -1, 15, 15, -1, -1, -1, -1, 15, -17, 15, -1, -1, 15, 15, 15, -17,
 -1, -17, -17, -17, -1, -1, 15, -1, -17, 15, -1, -1, 15, -1, 15, -1, 15, -1,
 15, -17, -1, -1, 15, -17, -1, -17, -17, 15, 15, -1, -1, 15, -1, -1, -1, -1,
 -17, -1, -17, -17, -17, -1, -17, -1, -1, -1, -17, -1, -1, -1, -1, -1, -1}
```

We generate the output of the modulator (GMSK modulation)

```
ModOutput = Modulator[L][(Goldseqlist // Last) /. {0 -> -1}, NumberOfCurves → 2,
  ModulatingPulse → FiltPulse, SamplingInterval → T/4];
```

We check the output

```
ListPlot[{Re[ModOutput], Im[ModOutput]} // Transpose, PlotJoined -> True,
 AspectRatio -> 1]
```



Filtered pulse

- Graphics -

Now we try to test the modulated sequence using the

Building a receiver based on pulse length

The number of samples per chip is equal to

$$\frac{3}{26000000} / T$$

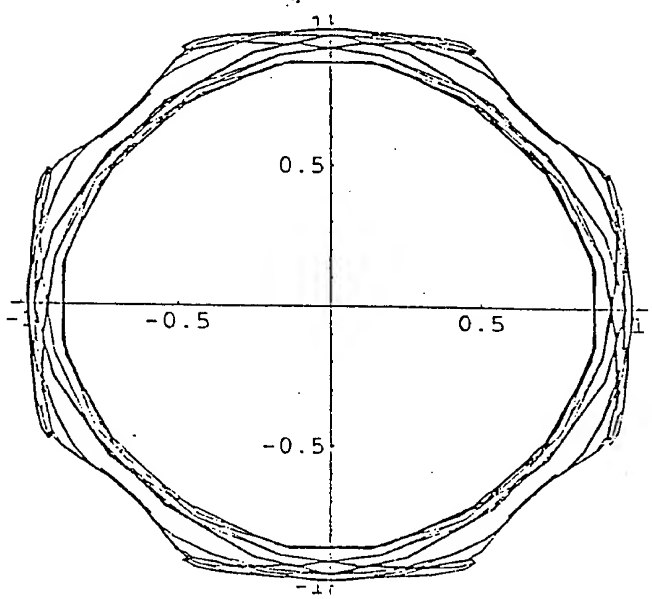
$$\frac{1}{32}$$

A check:- comparison of the received signal with what was being sent

```
(Receiver[L][ModOutput, StartingQuadrant -> 0,
    SamplingInterval → T/4, ModulatingPulse → FiltPulse] // Drop[#, 4]& ) -
  ((Goldseqlist // Last // Drop[#, -4]&) /. {0 -> -1})

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

We have successfully demodulated the bitstream

```
ModOutputOpt = Modulator[L][(Goldseqlist // Last) /. {0 -> -1},
  NumberOfCurves → 2, ModulatingPulse → OptPulseScaled, SamplingInterval → T/4];

ListPlot[{Re[ModOutputOpt], Im[ModOutputOpt]} // Transpose, PlotJoined -> True,
 AspectRatio -> 1]
```



An alternative modulator output

- preferred pulse shape

```
- Graphics -
```

Another check:-

```
(Receiver[L][ModOutputOpt, StartingQuadrant -> 0,
    SamplingInterval → T/4, ModulatingPulse → OptPulseScaled] // Drop[#, 4]& ) -
  ((Goldseqlist // Last // Drop[#, -4]&) /. {0 -> -1})

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

CDMA OPERATION

As an example given a symbol stream e.g. {1,1,-1,-1, ...} consisting of -1 and 1, the following function demonstrates the encoding process.

```
CDMAEncode[BiPolarBitSeq_, GoldSeq_] :=
  Module[{x1, x2, x3},
  x1 = GoldSeq /. {0 → -1};
  Map[x1 # &, BiPolarBitSeq] // Flatten]
```

Simulation of Encoding at transmitter

```
CDMAEncodedSeq = CDMAEncode[{-1, 1, 1, -1}, Goldseqlist // Last];
```

CDMA decoding of single Symbol

The modular output associated with {1}

```
ModOutputPlusOne = Modulator[L][CDMAEncode[{1}, (Goldseqlist // Last) /. {0 -> -1}],
  NumberOfCurves → 2, ModulatingPulse → OptPulseScaled, SamplingInterval → T / 4];
```

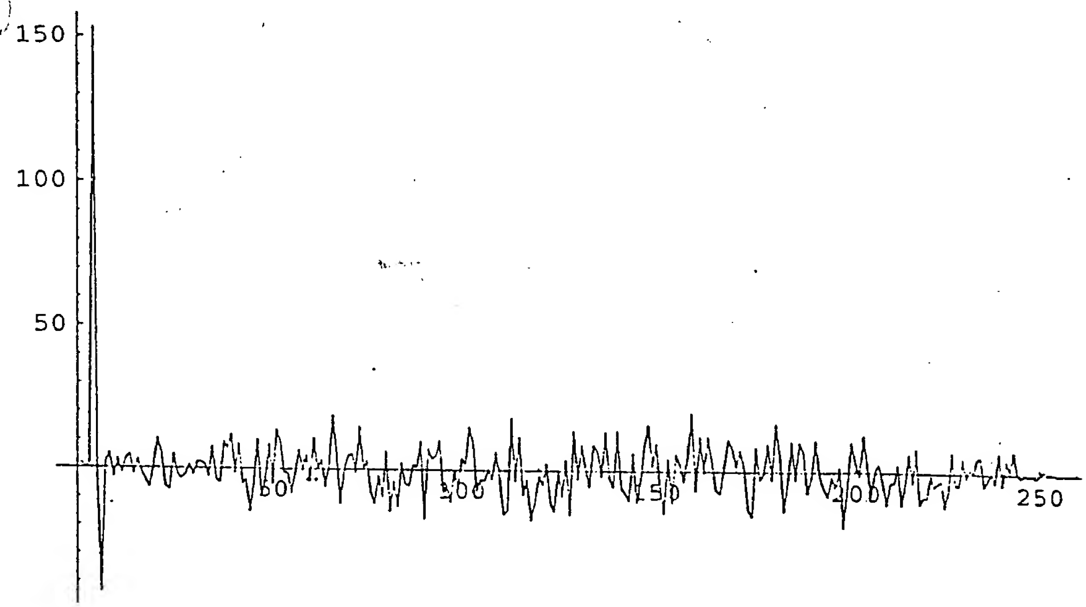
This is a primitive decoder built to study the autocorrelation. This will help in decoding

```
Take[ModOutputPlusOne, 10]

{0.691379 + 0.510132 I, 0.431257 + 0.748432 I, 0.130196 + 0.863809 I,
 -0.183585 + 0.853405 I, -0.510127 + 0.69136 I, -0.748423 + 0.431231 I,
 -0.863782 + 0.130145 I, -0.853326 - 0.183682 I, -0.69115 - 0.510321 I,
 -0.430757 - 0.74887 I}

PrimitiveCDMAReceiver[ModOutput_, GoldSeq_, Sample_, OverSampling_] :=
 Module[{x1, x2State, x3Update, x4, x5State, x6},
  x1 = Partition[ModOutput, OverSampling] // Transpose // #[[Sample]]&;
  x2State = GoldSeq /. {0 -> -1};
   x5State = Table[(-1)^Mod[i, 2], {i, 0, Length[x2State] - 1}];
  x3Update := Module[{},
    x2State = RotateRight[x2State]; x4 = FoldList[Plus, 0, x2State] // Rest // I^#&;
   x5State = RotateRight[x5State]; x1 (x5State x4) // Apply[Plus, #]&];
Table[x3Update, {i, 1, Length[GoldSeq]}]]
```

We make it more efficient

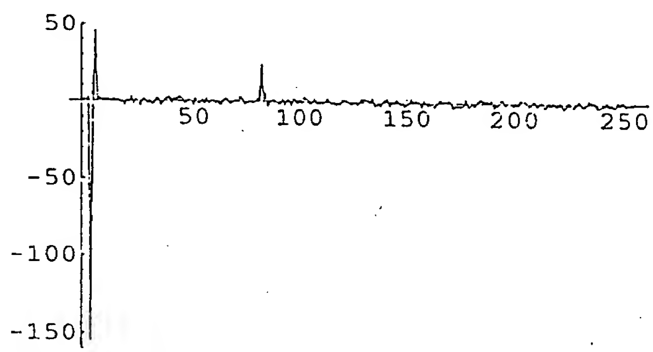
```
PrimitiveCDMAReceiver2[ModOutput_, GoldSeq_, Sample_, OverSampling_] :=
 Module[{x1, x2State, x3Update, x4, x5State, x6},
  x1 = Partition[ModOutput, OverSampling] // Transpose // #[[Sample]]&;
  x2State = GoldSeq /. {0 -> -1};
   x5State = Table[(1)^Mod[i, 2], {i, 0, Length[x2State] - 1}];
  x3Update := Module[{},
    x2State = RotateRight[x2State]; x4 = FoldList[Plus, 0, x2State] // Rest // I^-#&;
   x5State = RotateRight[x5State]; x1 (x5State x4) // Apply[Plus, #]&];
Table[x3Update, {i, 1, Length[GoldSeq]}]]

Tom = PrimitiveCDMAReceiver[ModOutputPlusOne, (Goldseqlist // Last), 1, 4];
```

```
ListPlot[Tom // Re, PlotJoined -> True, PlotRange -> All]
```



- Graphics -

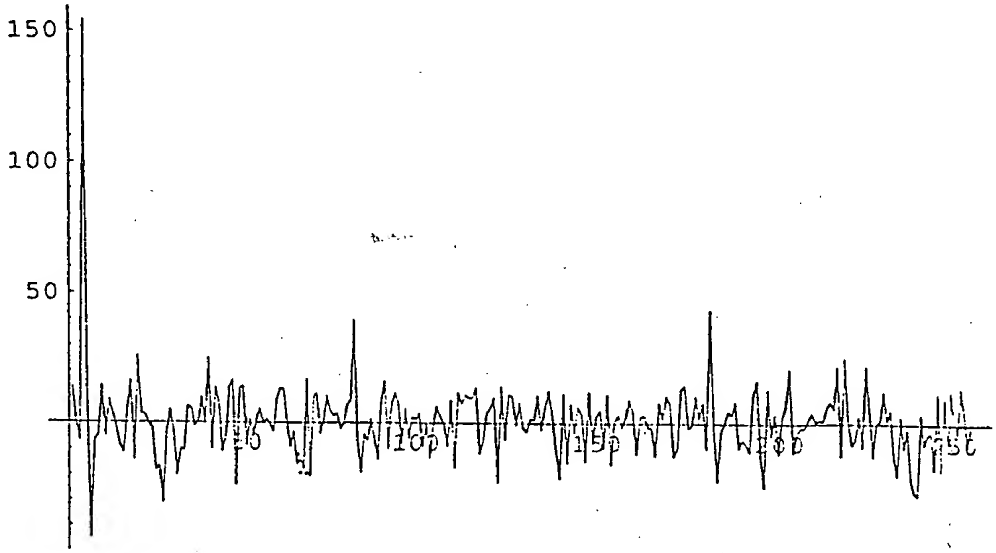
```
Tom = PrimitiveCDMAReceiver2[ModOutputPlusOne, (Goldseqlist // Last), 1, 4];

ListPlot[Tom // Re, PlotJoined -> True, PlotRange -> All]
```



- Graphics -

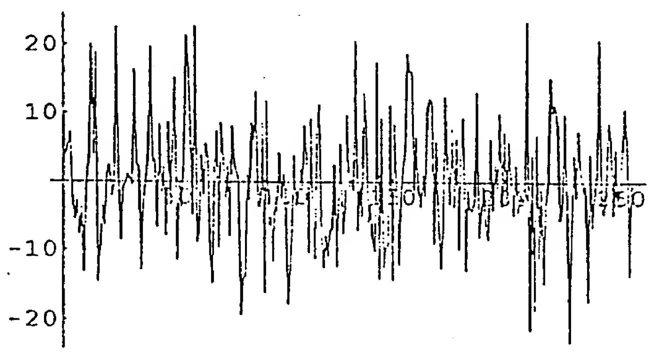
```
Tom3 = PrimitiveCDMAReceiver2[ModOutputPlusOne3, (Goldseqlist // #[[3]]&), 1, 4];
```

```
ListPlot[Tom3 // Re, PlotJoined -> True, PlotRange -> All]
```



- Graphics -

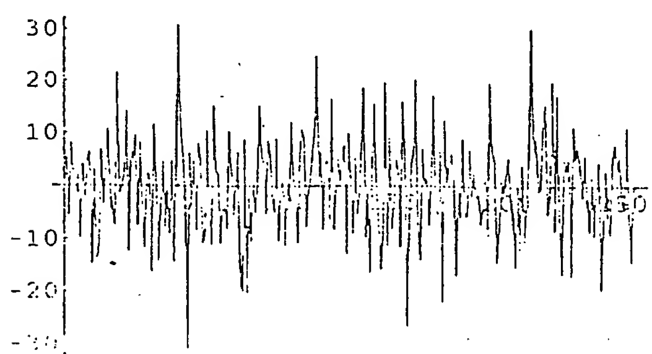
```
Tom4 = PrimitiveCDMAReceiver2[ModOutputPlusOne3, (Goldseqlist // #[[4]]&), 1, 4];
ListPlot[Tom4 // Re, PlotJoined -> True, PlotRange -> All]
```



'Wrong' Gold Code

- Graphics -

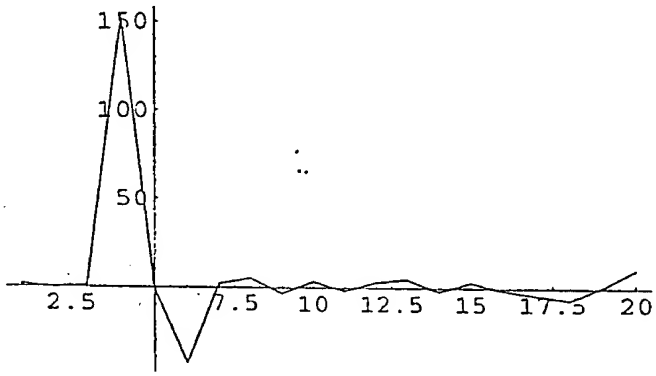
```
Tom5 = PrimitiveCDMAReceiver[ModOutputPlusOne3, (Goldseqlist // #[[4]]&), 1, 4];
ListPlot[Tom5 // Re, PlotJoined -> True, PlotRange -> All]
```



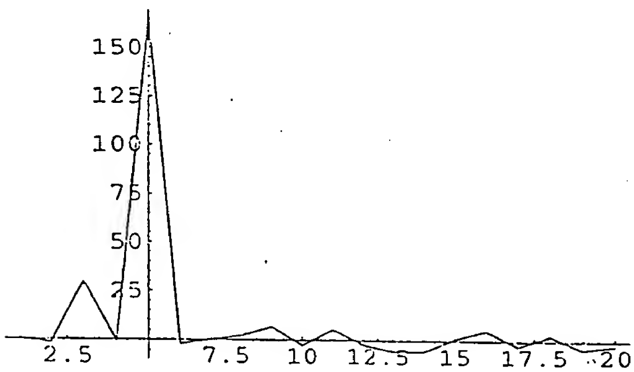
'Wrong' Gold Code

- Graphics -

```
Length[Tom]

255

Take [Tom, 20]

{1.75501 + 0.329995 I, 0.0343421 - 1.58366 I, 1.39349 + 29.6902 I, 153.017 - 0.957068 I,
 -0.718377 + 165.563 I, -43.6424 - 2.04777 I, 2.21031 + 0.256636 I, 5.392 + 2.18642 I,
 -3.13017 + 6.54167 I, 3.61193 - 2.5452 I, -1.68586 + 5.24864 I, 3.35077 - 1.96193 I,
 4.93408 - 5.10993 I, -1.90672 - 5.29083 I, 3.34843 + 0.914983 I, -1.24192 + 4.93512 I,
 -3.90572 - 2.50768 I, -6.70085 + 2.60649 I, 0.886488 - 4.00636 I, 10.5807 - 2.08322 I}

ListPlot[Tom // Re // Take[#, 20]&, PlotJoined -> True, PlotRange -> All]
```

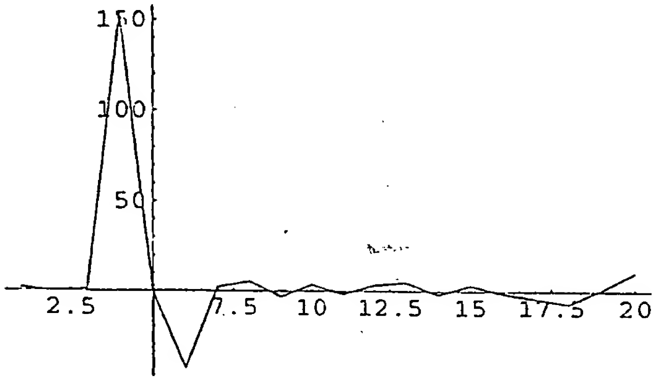



```
- Graphics -

ListPlot[Tom // Im // Take[#, 20]&, PlotJoined -> True, PlotRange -> All]
```



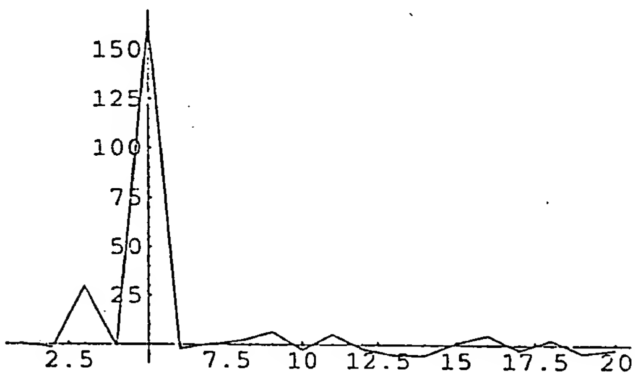

```
- Graphics -

Take[Tom, 10]

{1.75501 + 0.329995 I, 0.0343421 - 1.58366 I, 1.39349 + 29.6902 I, 153.017 - 0.957068 I,
 -0.718377 + 165.563 I, -43.6424 - 2.04777 I, 2.21031 + 0.256636 I, 5.392 + 2.18642 I,
 -3.13017 + 6.54167 I, 3.61193 - 2.5452 I}
```

```
ListPlot[Tom // Re // Take[#, 20]&, PlotJoined -> True, PlotRange -> All]
```



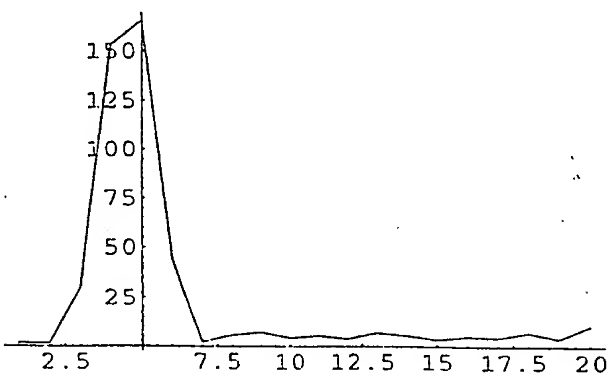
- Graphics -

```
ListPlot[Tom // Im // Take[#, 20]&, PlotJoined -> True, PlotRange -> All]
```



- Graphics -

```
ListPlot[Tom // Abs // Take[#, 20]&, PlotJoined -> True, PlotRange -> All]
```

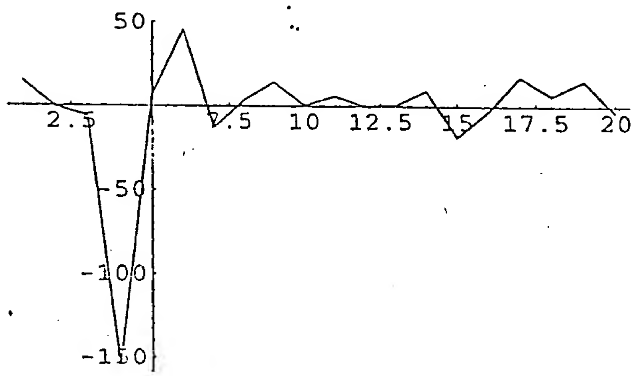


- Graphics -

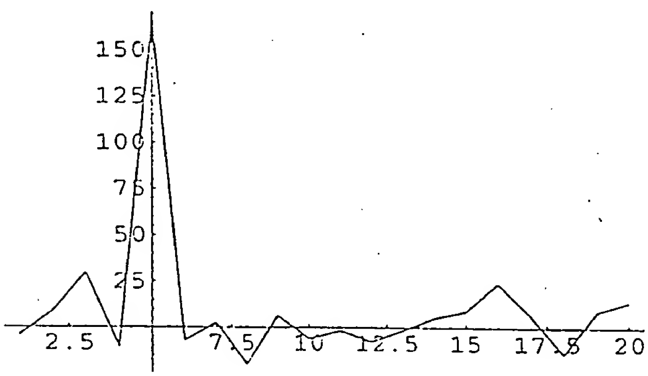
We try a less favourable sequence

```
ModOutputPlusOne3 = Modulator[L][CDMAEncode[{1}, Goldseqlist //#[[3]]&],
  NumberOfCurves -> 2, ModulatingPulse -> OptPulseScaled, SamplingInterval -> T/4];
```

```
PrimitiveCDMAReceiverMinus[ModOutput_, GoldSeq_, Sample_, OverSampling_] :=
 Module[{x1, x2State, x3Update, x4, x5State, x6},
  x1 = Partition[ModOutput, OverSampling] // Transpose // #[[Sample]]&;
  x2State = -(GoldSeq /. {0 -> -1});
   x5State = Table[(-1)^Mod[i, 2], {i, 0, Length[x2State] - 1}];
  x3Update := Module[{},
    x2State = RotateRight[x2State]; x4 = FoldList[Plus, 0, x2State] // Rest // I^#&;
   x5State = RotateRight[x5State]; x1 (x5State x4) // Apply[Plus, #]&];
Table[x3Update, {i, 1, Length[GoldSeq]}]]

Tom4 = PrimitiveCDMAReceiver[ModOutputPlusOne3, (Goldseqlist // #[[3]]&), 1, 4];

Take[Tom4, 10]

{14.7675 - 4.24542 I, -0.301874 + 8.61808 I, -5.92939 + 29.6038 I, -154.61 - 10.4754 I,
 6.64138 + 166.152 I, 45.2887 - 7.16233 I, -12.809 + 2.47796 I, 3.45948 - 19.7653 I,
 14.2097 + 6.33624 I, 0.585872 - 6.10244 I}

ListPlot[Tom4 // Re // Take[#, 20]&, PlotJoined -> True, PlotRange -> All]
```

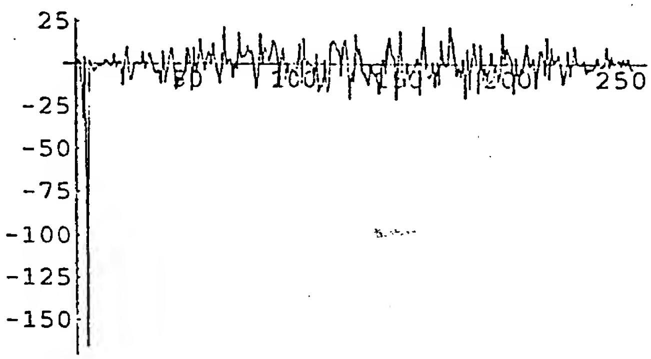



```
- Graphics -

ListPlot[Tom4 // Im // Take[#, 20]&, PlotJoined -> True, PlotRange -> All]
```

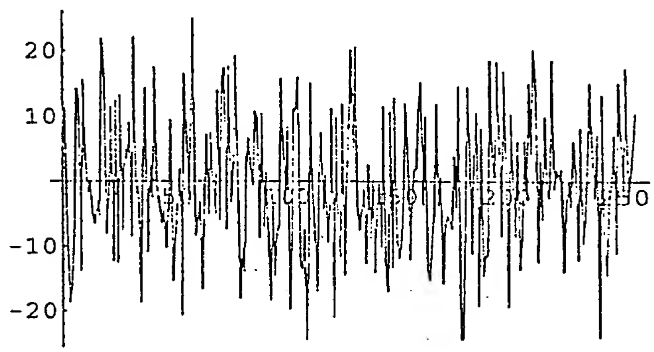



```
- Graphics -

ModOutputMinusOne =
 Modulator[L][CDMAEncode[{-1}, (Goldseqlist // Last) /. {0 -> -1}],
  NumberOfCurves → 2, ModulatingPulse → OptPulseScaled, SamplingInterval → T/4];

TomM1 = PrimitiveCDMAReceiverMinus[ModOutputMinusOne, (Goldseqlist // Last), 1, 4];
```

```
ListPlot[TomM1 // Im, PlotJoined -> True, PlotRange -> All]
```




```
- Graphics -

TomM1 = PrimitiveCDMAReceiver[ModOutputMinusOne, (Goldseqlist // Last), 1, 4];

ListPlot[TomM1 // Im, PlotJoined -> True, PlotRange -> All]
```



Normal receiver for +1 used to try to detect a -1.

```
- Graphics -

Length[Tom]

255

Take [Tom, 20]

{1.75501 + 0.329995 I, 0.0343421 - 1.58366 I, 1.39349 + 29.6902 I, 153.017 - 0.957068 I,
 -0.718377 + 165.563 I, -43.6424 - 2.04777 I, 2.21031 + 0.256636 I, 5.392 + 2.18642 I,
 -3.13017 + 6.54167 I, 3.61193 - 2.5452 I, -1.68586 + 5.24864 I, 3.35977 - 1.96193 I,
 4.93408 - 5.10993 I, -1.90672 - 5.29083 I, 3.34843 + 0.914983 I, -1.24192 + 4.93512 I,
 -3.90572 - 2.50768 I, -6.70085 + 2.60649 I, 0.886488 - 4.00636 I, 10.5807 - 2.08322 I}
```

Checking the correlation properties of the Training Sequence in GSM

```
GSM = {0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1}

{0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1}

GSMseq = {-1, -1, 1, -1, -1, 1, -1, 1, 1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1,
   -1, -1, 1, -1, 1, 1, 1}

{-1, -1, 1, -1, -1, 1, -1, 1, 1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1,
 1, -1, 1, 1, 1}
```

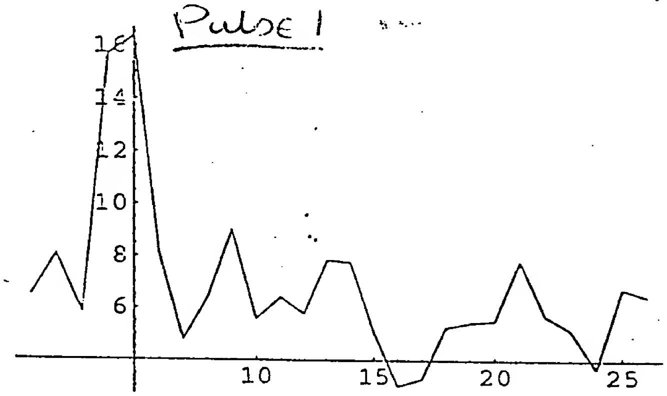
GSMseq = GSM training sequence used as despreading code.

```
AutocorrelationSequence[GSMseq]
```

– Checking correlation properties of GSM train sequence.

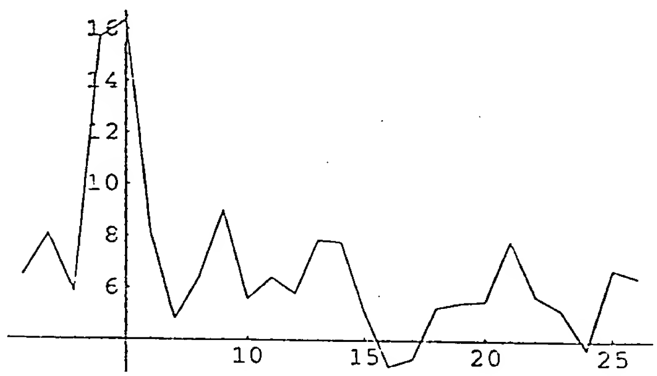
```
{26, -2, -2, 2, -2, -2, -2, 6, -10, 2, 10, -2, -2, -2, -2, -2, 10, 2, -10, 6,
 -2, -2, -2, 2, -2, -2}

ModOutputGSM = Modulator[L][GSMseq, NumberOfCurves → 2,
  ModulatingPulse → OptPulseScaled, SamplingInterval → T/4];

TomGSM1 = PrimitiveCDMAReceiver[ModOutputGSM, GSMseq, 1, 4];

ListPlot[TomGSM1 // Abs, PlotJoined -> True, PlotRange -> All]
```

Pulse 1



Transformation 1

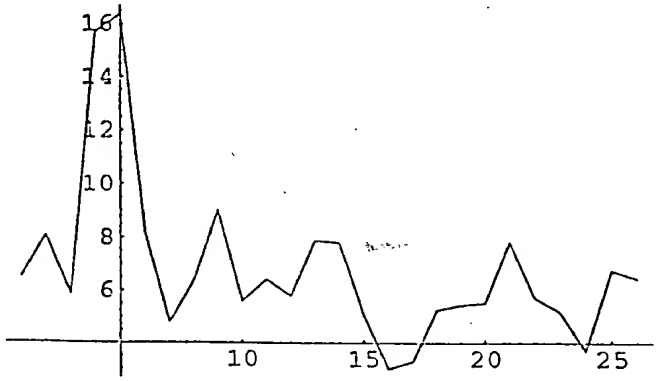
```
- Graphics -

PrimitiveCDMAReceiver2[ModOutputGSM, GSMseq, 1, 4];

ListPlot[%90 // Abs, PlotJoined -> True, PlotRange -> All]
```



Transformation 1b.

```
- Graphics -

PrimitiveCDMAReceiver[ModOutputGSM, GSMseq, 1, 4];
```

```
ListPlot[%92 // Abs, PlotJoined -> True, PlotRange -> All]
```

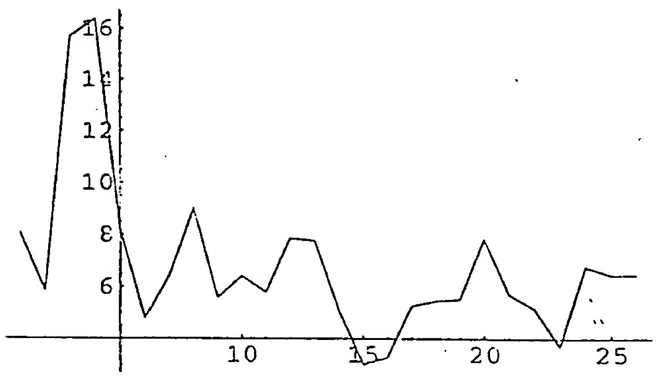


- Graphics -

```
PrimitiveCDMAReceiverGSM2Pulse[ModOutput_, GoldSeq_, Sample_, OverSampling_] :=
 Module[{x1, x2State, x3Update, x4, x5State, x5, x6},
  x1 = Partition[ModOutput, OverSampling] // Transpose // #[[Sample]]&;
  x2State = GoldSeq /. {0 -> -1};
   x5State = Table[(-1)^Mod[i, 2], {i, 0, Length[x2State] - 1}];
  x3Update := Module[{}, x2State = RotateRight[x2State];
    x4 = FoldList[Plus, 0, x2State] // Rest;
      x5 = Join[{1}, x2State // Drop[#, -1]&];
      x6 = FoldList[Plus, 0, x5] // Rest // I^#&;
   x5State = RotateRight[x5State]; x1 (x5State x6) // Apply[Plus, #]&];
Table[x3Update, {i, 1, Length[GoldSeq]}]]

TomGSMSecondP = PrimitiveCDMAReceiverGSM2Pulse[ModOutputGSM, GSMseq, 1, 4];
```

Pulse 2.

```
ListPlot[TomGSMSecondP // Abs, PlotJoined -> True, PlotRange -> All]
```

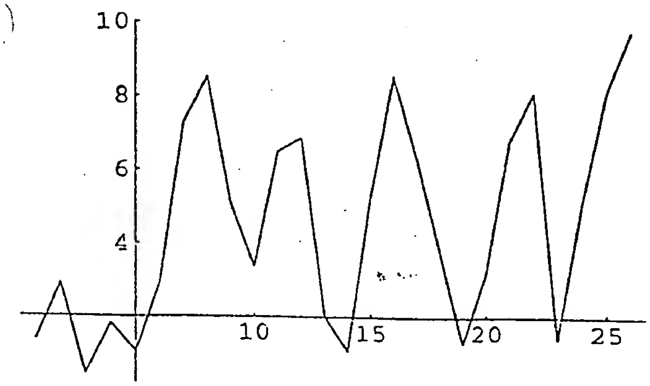


Transformation 3

- Graphics -

```
PrimitiveCDMAReceiverGSM2PulseEfficient[
  ModOutput_, GoldSeq_, Sample_, OverSampling_] :=
 Module[{x1, x2State, x3Update, x4, x5State, x5, x6},
  x1 = Partition[ModOutput, OverSampling] // Transpose // #[[Sample]]&;
  x2State = GoldSeq /. {0 -> -1};
   x5State = Table[(1)^Mod[i, 2], {i, 0, Length[x2State] - 1}];
  x3Update := Module[{}, x2State = RotateRight[x2State];
    x4 = FoldList[Plus, 0, x2State] // Rest;
      x5 = Join[{1}, x2State // Drop[#, -1]&];
      x6 = FoldList[Plus, 0, x5] // Rest // I^#&;
   x5State = RotateRight[x5State]; x1 (x5State x6) // Apply[Plus, #]&];
Table[x3Update, {i, 1, Length[GoldSeq]}]]

TomGSMSecondPEff2 =
 PrimitiveCDMAReceiverGSM2PulseEfficient[ModOutputGSM, GSMseq, 1, 4];
```

```
ListPlot[TomGSMSecondPEff2 // Abs, PlotJoined -> True, PlotRange -> All]
```



Transformation 3b

```
- Graphics -
```

▫ CDMA Decoding of Several Symbols with Training sequence Receiver

First we generate the modulator output. For simplicity we will use a very short training sequence . Let the training sequence one of GSM training sequences Let the guard sequences be {1,1,1} . Let the data symbols be generated by a random

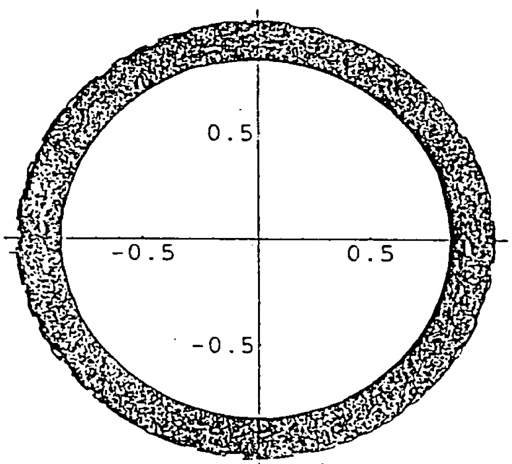
```
data1 = {1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0};

data2 = {0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1};

guard = {1, 1, 1}

{1, 1, 1}

training = {0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1};
```

The GSM Training sequence is {0,0,1,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,1,0,0,1,0,,1,1,1}. In fact any short m–sequence can be used to characterise the output.

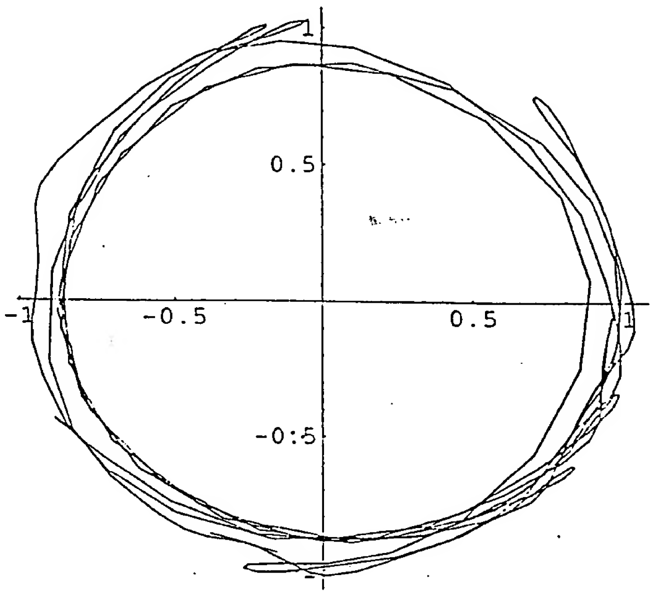
Only at this point that we differentially encode using the gsm scheme

```
frame = Join[guard, data1, training, data2, guard]

General::spell1 :
 Possible spelling error: new symbol name "frame" is similar to existing symbol "Frame".

{1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0,
 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1,
 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1}

GSMDiffEncodedFrame[frame_] := Module[{x1, x2, x3}, x1 = Partition[frame, 2, 1];
  x2 = Map[Mod[#[[1]] + #[[2]], 2]&, x1] // Join[{frame[[1]]}, #]&;
 x3 = x2 /. {0 -> 1, 1 -> -1}]

General::spell1 :
 Possible spelling error: new symbol name "frame" is similar to existing symbol "Frame".

frameEncoded = GSMDiffEncodedFrame[frame]

{-1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, -1, -1, 1, 1,
 -1, -1, 1, -1, -1, -1, 1, 1, -1, 1, 1, 1, -1, -1, 1, 1, -1, -1, 1, -1, -1, -1,
 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, 1, -1, 1, 1, 1, 1}

ModOutputFrame3 = Modulator[L][CDMAEncode[frameEncoded, GoldSeqList // Last],
  NumberOfCurves -> 2, ModulatingPulse -> OptPulseScaled, SamplingInterval -> T / 4];
```

```
Carrfreq001 = Table[E^(I 2 Pi 0.001 j) // N, {j, 0, Length[ModOutputFrame3] - 1}];

Carr001 = ModOutputFrame3 Carrfreq001;

Save["Carr001.m", Carr001];

Carrfreq0005 = Table[E^(I 2 Pi 0.0005 j) // N, {j, 0, Length[ModOutputFrame3] - 1}];

Carr0005 = ModOutputFrame3 Carrfreq0005;

Save["Carr0005.m", Carr0005];

Carrfreq002 = Table[E^(I 2 Pi 0.002 j) // N, {j, 0, Length[ModOutputFrame3] - 1}];

Carr002 = ModOutputFrame3 Carrfreq002;

Save["Carr002.m", Carr002];

ModOutputUnEncodedFrame3 =
 Modulator[L][CDMAEncode[frame /. 0 -> -1, Goldseqlist // #[[3]]&],
  NumberOfCurves → 2, ModulatingPulse → OptPulseScaled, SamplingInterval → T/4];

Save["ModOutputUnEncodedFrameGSMLike3.m", ModOutputFrame]

<< ModOutputFrameEncodedGSMLike.m;

Length[ModOutputFrame]

73440

AFC0005 = Take[Carr0005, {50, 7300}];

AFC0005 // {Re[#], Im[#]}& // Transpose //
 ListPlot[#, PlotJoined -> True, PlotRange -> All, AspectRatio -> 1]&;
```



```
AFC0005 // Take[#, 200]& // {Re[#], Im[#]}& // Transpose //
 ListPlot[#, PlotJoined -> True, PlotRange -> All, AspectRatio -> 1]&;
```



In the PrimitiveCDMA receiver we need to specify the sample. In the CDMA synchroniser we discover the sample.

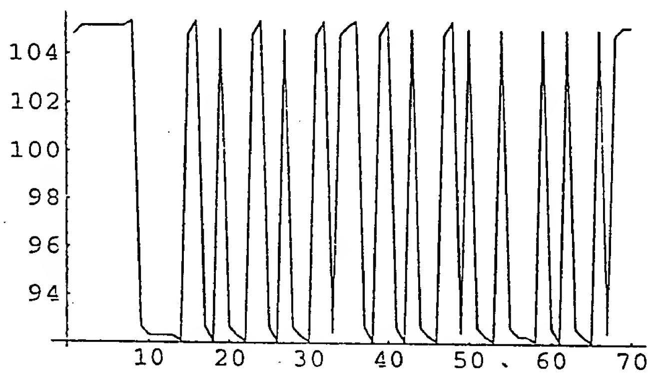
```
CDMACoarseSynchroniserNew[ModOutput_, GoldSeq_, Threshold_, OverSampling_] :=
 Module[{x1, x2, x3, x4Plus, x4Minus, x5State,
   x6Count, x6MaxCorr, seq, x7Update, x8, x9, x10, x11, x12, x13},
  x1 = ModOutput;
  x2 = GoldSeq /. 0 -> -1;
  x3 = CodingTransformNew[L][GoldSeq];
  x4Plus = x3[[1]] ;
  x4Minus = x3[[2]];
  x5State = Take[x1, (Length[x2] + 1) OverSampling];
  x6Count = 1;
  x6MaxCorr = 0;
  seq = Drop[x1, OverSampling (Length[x2] + 1)];
 x7Update := Module[{},
x8 = Partition[x5State, OverSampling] // Transpose;
x9 = Map[{Drop[#, -1] . x4Plus, Drop[#, 1] . x4Minus}&, x8];
x10 = Map[
     {Abs[Im[#[[1]]]], Abs[Re[#[[1]]]], Abs[Re[#[[2]]]], Abs[Im[#[[2]]]] }&, x9];
x11 = If[Max[x10] > Threshold, Throw[{x10, x6Count, True}],
     {x6Count, x10, x6MaxCorr = Max[x6MaxCorr, x10], False}];
    x6Count = x6Count + 1;
    x5State = Join[Drop[x5State, OverSampling], Take[seq, OverSampling]];
    seq = Drop[seq, OverSampling];
    x11 ];
x12 = Catch[Table[x7Update, {i, 1, Length[seq] / OverSampling }]];
 x13 = If[Last[x12] === True,
    CDMAFineSynchroniser[ModOutput, x12[[2]], x3, OverSampling],
    {"Failed to Coarse Synchronise", False}]]

Tom4 =
 CDMACoarseSynchroniserNew[AFC0005 // Drop[#, 250 4]&, Goldseqlist // Last , 50, 4]

DeModulator[
 {{{16.252 - 103.911 I, 9.19889 + 6.93038 I, -3.61972 + 15.2763 I, 13.0046 - 9.41072 I},
   {-102.686 - 22.7446 I, 7.49431 - 8.74558 I, 15.0189 + 4.57178 I, -8.57558 - 13.5698 I},
   {-29.1474 + 101.055 I, -8.25775 - 8.02866 I, 5.50581 - 14.7022 I, -14.0815 + 7.70661 I},
   {99.0255 + 35.4352 I, -8.53133 + 7.73733 I, -14.3275 - 6.4181 I, 6.80721 + 14.5376 I},
   {41.5832 - 96.6051 I, 7.18638 + 9.00033 I, -7.30507 + 13.8962 I, 14.9364 - 5.88096 I}},
  {{24.6533 - 96.8492 I, 10.1266 + 6.85283 I, 1.46802 - 2.37691 I, 12.5087 - 10.488 I},
   {-95.1101 - 30.6858 I, 7.47516 - 9.6763 I,
    -2.28004 - 1.61437 I, -9.68188 - 13.1425 I}, {-36.5973 + 92.9936 I,
    9.18784 - 8.06799 I, -1.75435 + 2.17418 I, -13.7245 + 8.83754 I},
   {90.5141 + 42.3543 I, -8.62898 + 8.66312 I, 2.05973 + 1.8874 I, 7.55834 + 14.2523 I},
   {47.9641 - 87.6755 I, 8.1042 + 9.15591 I, 2.01301 - 1.93715 I, 14.7239 - 7.04772 I}}}]
```

```
Tom0005 =
 CDMACoarseSynchroniserNew[Carr0005 // Drop[#, 250 4]&, Goldseqlist // Last , 100, 4];

CDMAFineSynchroniser[ModOutput_,
  ThresholdCorrelatioCount_, CorrelatingSeq_, OverSampling_] :=
   Module[{x1, x2, x3, x4, x5, x6, seq, x7Update, x8First,
   x8Second, x9First, x9Second, x10First, x10Second, x11, x12, x13, x14},
  x1 = If[ThresholdCorrelatioCount > 3,
    ThresholdCorrelatioCount - 3, ThresholdCorrelatioCount];
  x2 = CorrelatingSeq;
  x3 = Drop[ModOutput, x1 OverSampling];
 x4 = CDMAPositionFinder[x3, x2, OverSampling];
  x5 = CDMAPositionFinder[Drop[x3, Length[x2] OverSampling] , x2, OverSampling];
   x6 = CDMAPositionFinder[Drop[x3, 2 Length[x2] OverSampling], x2, OverSampling];
  x7 = PositionAverager[{x4[[1]], x5[[1]], x6[[1]]}];
  x8First = Drop[x3, (x7[[1]] - 1) OverSampling] //
      Partition[#, OverSampling]& // Transpose // #[[x7[[2]]]]&;
   x8Second = Drop[x3, x7[[1]] OverSampling] //
      Partition[#, OverSampling]& // Transpose // #[[x7[[2]]]]&;
 x9First = x8First // Partition[#, Length[x2[[1]]]]&;
  x10First = Map[Function[x, Map[x . #&, x2]], x9First];
    x9Second = x8Second // Partition[#, Length[x2[[1]]]]&;
  x10Second = Map[Function[x, Map[x . #&, x2]], x9Second] ;
  DeModulator[{x10First, x10Second}]  ]

CDMAPositionFinder[ModOutput_, CorrelatingSeq_, OverSampling_] :=
   Module[{x5State, x6Count, seq, x7Update, x8, x9, x10, x11, x12, x13},
 x5State = Take[ModOutput, (Length[CorrelatingSeq[[1]]] + 1) OverSampling];
  x6Count = 1;
  seq = Drop[ModOutput, OverSampling (Length[CorrelatingSeq[[1]]] + 1)];
  x7Update := Module[{},
x8 = Partition[x5State, OverSampling] // Transpose;
x9 =  Map[{Drop[#, -1] . CorrelatingSeq[[1]], Drop[#, -1] . CorrelatingSeq[[2]],
        Drop[#, 1] . CorrelatingSeq[[1]], Drop[#, 1] . CorrelatingSeq[[2]]}&, x8];
     x6Count = x6Count + 1;
     x5State = Join[Drop[x5State, OverSampling], Take[seq , OverSampling]];
     seq = Drop[seq, OverSampling];
     x9 ];
x10 =  Table[x7Update, {i, 1, 10}];
x11 =  MapIndexed[Max[{Abs[Re[#]], Abs[Im[#]]}]&, x10, {3}];
x12 = MapIndexed[Apply[Plus, #]&, x11, {2}];
  x13 = Position[x12, Max[x12]] ]
```

We need to define a position averager. We for now just take the first element

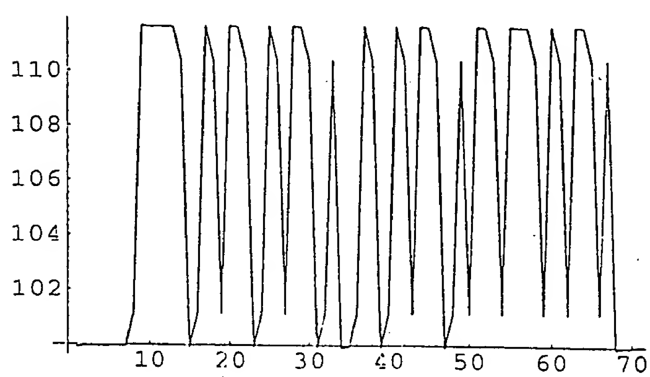
```
PositionAverager[PositionList_] := First[PositionList]

Tom4 =
 CDMACoarseSynchroniserNew[TestData // Drop[#, 250 4]&, Goldseqlist // Last , 100, 4]

DeModulator[
 {{{-157.056 + 0.691379 I, 7.37139 - 18.8305 I, 41.1322 + 1.94693 I, -16.2843 - 23.4715 I},
   {-0.691379 - 157.056 I, 18.8305 + 7.37139 I, -1.94693 + 41.1322 I, 23.4715 - 16.2843 I},
   {157.056 - 0.691379 I, -7.37139 + 18.8305 I, -41.1322 - 1.94693 I, 16.2843 + 23.4715 I},
   {0.691379 + 157.056 I, -18.8305 - 7.37139 I,
    1.94693 - 41.1322 I, -23.4715 + 16.2843 I}, {-157.056 + 0.691379 I,
    7.37139 - 18.8305 I, 41.1322 + 1.94693 I, -16.2843 - 23.4715 I}},
  {{-169.734 - 0.510132 I, 6.8871 - 20.5026 I, 6.24607 + 1.47947 I, -17.4944 - 21.5101 I},
   {0.510132 - 169.734 I, 20.5026 + 6.8871 I, -1.47947 + 6.24607 I, 21.5101 - 17.4944 I},
   {169.734 + 0.510132 I, -6.8871 + 20.5026 I, -6.24607 - 1.47947 I, 17.4944 + 21.5101 I},
   {-0.510132 + 169.734 I, -20.5026 - 6.8871 I, 1.47947 - 6.24607 I, -21.5101 + 17.4944 I},
   {-169.734 - 0.510132 I, 6.8871 - 20.5026 I, 6.24607 + 1.47947 I,
    -17.4944 - 21.5101 I}}}]
```

```
Tom4[[1]] // Transpose

{{{-157.056 + 0.691379 I, 7.37139 - 18.8305 I, 41.1322 + 1.94693 I, -16.2843 - 23.4715 I},
  {-169.734 - 0.510132 I, 6.8871 - 20.5026 I, 6.24607 + 1.47947 I, -17.4944 - 21.5101 I}},
 {{-0.691379 - 157.056 I, 18.8305 + 7.37139 I, -1.94693 + 41.1322 I, 23.4715 - 16.2843 I},
  {0.510132 - 169.734 I, 20.5026 + 6.8871 I, -1.47947 + 6.24607 I, 21.5101 - 17.4944 I}},
 {{157.056 - 0.691379 I, -7.37139 + 18.8305 I, -41.1322 - 1.94693 I, 16.2843 + 23.4715 I},
  {169.734 + 0.510132 I, -6.8871 + 20.5026 I, -6.24607 - 1.47947 I, 17.4944 + 21.5101 I}},
 {{0.691379 + 157.056 I, -18.8305 - 7.37139 I, 1.94693 - 41.1322 I, -23.4715 + 16.2843 I},
  {-0.510132 + 169.734 I, -20.5026 - 6.8871 I, 1.47947 - 6.24607 I, -21.5101 + 17.4944 I}},
 {{-157.056 + 0.691379 I, 7.37139 - 18.8305 I, 41.1322 + 1.94693 I, -16.2843 - 23.4715 I},
  {-169.734 - 0.510132 I, 6.8871 - 20.5026 I, 6.24607 + 1.47947 I, -17.4944 - 21.5101 I}}}

Take[Tom5[[1]] // Transpose, {8, 10}]

{{{157.25 - 0.929033 I, -7.57128 + 19.0518 I, -40.9109 - 1.74704 I, 16.522 + 23.6653 I},
  {169.813 - 0.714545 I, -7.42576 + 21.3328 I, -5.4158 - 0.940778 I, 18.7191 + 21.5891 I}},
 {{19.1272 - 6.93509 I, -1.02745 + 156.682 I, 23.9098 + 15.9863 I, -1.57069 - 41.4696 I},
  {20.5326 - 6.8667 I, 0.478141 + 169.715 I, 21.5305 + 17.4644 I, -1.46045 - 6.27806 I}},
 {{-7.37139 - 18.8305 I, 157.056 + 0.691379 I, 16.2843 - 23.4715 I, -41.1322 + 1.94693 I},
  {-6.8871 - 20.5026 I, 169.734 - 0.510132 I, 17.4944 - 21.5101 I, -6.24607 + 1.47947 I}}}

Tom6[[1, 1]] // Transpose // {#[[1]], #[[2]]}& // Transpose // Map[Max, Abs[#]]& //
 ListPlot[#, PlotJoined -> True]&
```



- Graphics -

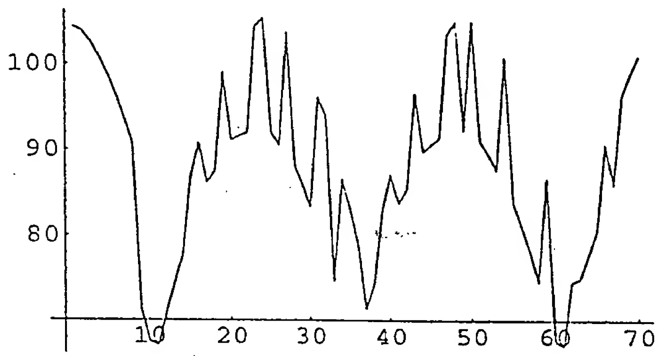
```
Tom6[[1, 2]] // Transpose // {#[[1]], #[[2]]}& // Transpose // Map[Max, Abs[#]]& //
 ListPlot[#, PlotJoined -> True]&
```



- Graphics -

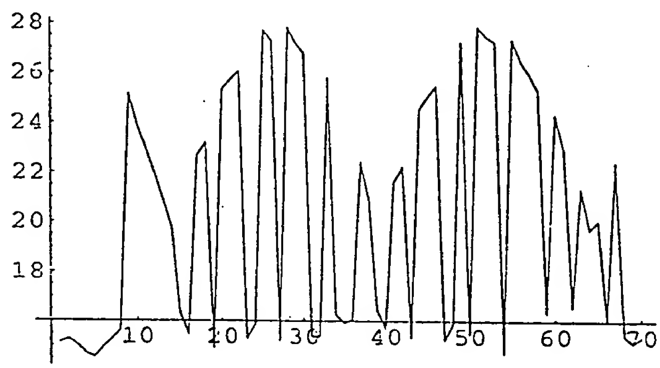
```
Map[Abs, Tom6[[1, 2]]]

{{99.9069, 12.1968, 2.82297, 16.3585}, {99.9377, 12.2274, 2.7937, 16.3237},
 {99.9377, 12.2274, 2.7937, 16.3237}, {99.9377, 12.2274, 2.7937, 16.3237},
 {99.9377, 12.2274, 2.7937, 16.3237}, {99.9377, 12.2274, 2.7937, 16.3237},
 {99.9377, 12.2274, 2.7937, 16.3237}, {101.16, 12.0496, 2.71211, 17.0171},
 {10.9504, 111.653, 17.2325, 16.2024}, {10.9205, 111.621, 17.2666, 16.1692},
 {10.9205, 111.621, 17.2666, 16.1692}, {10.9205, 111.621, 17.2666, 16.1692},
 {10.9205, 111.621, 17.2666, 16.1692}, {11.1111, 110.39, 16.592, 16.2327},
 {99.9069, 12.1968, 2.82297, 16.3585}, {101.16, 12.0496, 2.71211, 17.0171},
 {10.9504, 111.653, 17.2325, 16.2024}, {11.1111, 110.39, 16.592, 16.2327},
 {101.129, 12.0176, 2.74759, 17.0524}, {10.9504, 111.653, 17.2325, 16.2024},
 {10.9205, 111.621, 17.2666, 16.1692}, {11.1111, 110.39, 16.592, 16.2327},
 {99.9069, 12.1968, 2.82297, 16.3585}, {101.16, 12.0496, 2.71211, 17.0171},
 {10.9504, 111.653, 17.2325, 16.2024}, {11.1111, 110.39, 16.592, 16.2327},
 {101.129, 12.0176, 2.74759, 17.0524}, {10.9504, 111.653, 17.2325, 16.2024},
 {10.9205, 111.621, 17.2666, 16.1692}, {11.1111, 110.39, 16.592, 16.2327},
 {99.9069, 12.1968, 2.82297, 16.3585}, {101.16, 12.0496, 2.71211, 17.0171},
 {11.1392, 110.422, 16.5587, 16.2668}, {99.9069, 12.1968, 2.82297, 16.3585},
 {99.9377, 12.2274, 2.7937, 16.3237}, {101.16, 12.0496, 2.71211, 17.0171},
 {10.9504, 111.653, 17.2325, 16.2024}, {11.1111, 110.39, 16.592, 16.2327},
 {99.9069, 12.1968, 2.82297, 16.3585}, {101.16, 12.0496, 2.71211, 17.0171},
 {10.9504, 111.653, 17.2325, 16.2024}, {11.1111, 110.39, 16.592, 16.2327},
 {101.129, 12.0176, 2.74759, 17.0524}, {10.9504, 111.653, 17.2325, 16.2024},
 {10.9205, 111.621, 17.2666, 16.1692}, {11.1111, 110.39, 16.592, 16.2327},
 {99.9069, 12.1968, 2.82297, 16.3585}, {101.16, 12.0496, 2.71211, 17.0171},
 {11.1392, 110.422, 16.5587, 16.2668}, {101.129, 12.0176, 2.74759, 17.0524},
 {10.9504, 111.653, 17.2325, 16.2024}, {10.9205, 111.621, 17.2666, 16.1692},
 {11.1111, 110.39, 16.592, 16.2327}, {101.129, 12.0176, 2.74759, 17.0524},
 {10.9504, 111.653, 17.2325, 16.2024}, {10.9205, 111.621, 17.2666, 16.1692},
 {10.9205, 111.621, 17.2666, 16.1692}, {11.1111, 110.39, 16.592, 16.2327},
 {101.129, 12.0176, 2.74759, 17.0524}, {10.9504, 111.653, 17.2325, 16.2024},
 {11.1111, 110.39, 16.592, 16.2327}, {101.129, 12.0176, 2.74759, 17.0524},
 {10.9504, 111.653, 17.2325, 16.2024}, {10.9205, 111.621, 17.2666, 16.1692},
 {11.1111, 110.39, 16.592, 16.2327}, {101.129, 12.0176, 2.74759, 17.0524},
 {11.1392, 110.422, 16.5587, 16.2668}, {99.9069, 12.1968, 2.82297, 16.3585},
 {99.9377, 12.2274, 2.7937, 16.3237}, {99.9377, 12.2274, 2.7937, 16.3237}}

Max0005 =
 Map[{Max[{Re[#[[1]]] // Abs, Im[#[[1]]] // Abs, Re[#[[2]]] // Abs, Im[#[[2]]] // Abs}],
    Max[{Re[#[[3]]] // Abs, Im[#[[3]]] // Abs, Re[#[[4]]] // Abs, Im[#[[4]]] // Abs}]}&,
  Tom6[[1, 1]]]

{{104.351, 15.1379}, {103.911, 15.2763}, {102.686, 15.0189},
 {101.055, 14.7022}, {99.0255, 14.5376}, {96.6051, 14.9364},
 {93.8035, 15.2762}, {90.7175, 15.6415}, {71.4359, 25.1654}, {67.5608, 23.8519},
 {67.0266, 22.9468}, {70.8801, 21.9511}, {74.4539, 20.8688}, {77.6444, 19.7766},
 {86.7967, 16.2927}, {90.7238, 15.4739}, {86.1537, 22.7114}, {87.5152, 23.2037},
 {98.9714, 14.9013}, {91.0278, 25.3506}, {91.5367, 25.7331}, {91.8871, 26.0709},
 {104.382, 15.2963}, {105.346, 15.8922}, {91.9518, 27.7219}, {90.4855, 27.2999},
 {103.704, 15.2097}, {88.1543, 27.846}, {85.9746, 27.2075}, {83.3983, 26.8136},
 {96.1447, 15.3897}, {93.9076, 15.3803}, {74.6965, 25.8353}, {85.612, 16.2707},
 {83.2288, 15.9297}, {79.0562, 16.0518}, {71.3823, 22.4556}, {74.383, 20.923},
 {82.9341, 16.4519}, {87.1763, 15.7147}, {83.839, 21.6473}, {85.5445, 22.25},
 {96.5637, 15.3294}, {89.7552, 24.5667}, {90.6053, 25.0753}, {91.3309, 25.5023},
 {103.567, 15.2284}, {104.931, 15.891}, {92.2522, 27.2697}, {104.992, 15.4722},
 {91.0436, 27.8733}, {89.5104, 27.4849}, {87.6329, 27.2718}, {100.78, 14.7073},
 {83.8748, 27.3795}, {81.0829, 26.501}, {78.152, 25.9901}, {74.6068, 25.3375},
 {86.6792, 16.3378}, {67.5876, 24.357}, {66.9937, 22.9638}, {74.5498, 16.5852},
 {74.9528, 21.37}, {77.7339, 19.7041}, {80.5993, 20.0831}, {90.616, 16.0026},
 {86.0108, 22.4409}, {96.2752, 15.4329}, {98.8878, 15.1306}, {100.942, 15.3636}}
```

```
ListPlot[Max0005 // Transpose // #[[1]]&, PlotJoined -> True]
```



- Graphics -

```
ListPlot[Max0005 // Transpose // #[[2]]&, PlotJoined -> True]
```



- Graphics -

```
Map[{Max[{Re[#[[1]]] // Abs, Im[#[[1]]] // Abs, Re[#[[2]]] // Abs, Im[#[[2]]] // Abs}],
   Max[{Re[#[[3]]] // Abs, Im[#[[3]]] // Abs, Re[#[[4]]] // Abs, Im[#[[4]]] // Abs}]}&,
 Tom5[[1, 2]]]

{{169.715, 21.5305}, {169.734, 21.5101}, {169.734, 21.5101},
 {169.734, 21.5101}, {169.734, 21.5101}, {169.734, 21.5101},
 {169.734, 21.5101}, {169.813, 21.5891}, {169.715, 21.5305}, {169.734, 21.5101},
 {169.734, 21.5101}, {169.734, 21.5101}, {169.734, 21.5101}, {169.813, 21.5891},
 {169.715, 21.5305}, {169.813, 21.5891}, {169.715, 21.5305}, {169.813, 21.5891},
 {169.794, 21.6095}, {169.715, 21.5305}, {169.734, 21.5101}, {169.813, 21.5891},
 {169.715, 21.5305}, {169.813, 21.5891}, {169.715, 21.5305}, {169.813, 21.5891},
 {169.794, 21.6095}, {169.715, 21.5305}, {169.734, 21.5101}, {169.813, 21.5891},
 {169.715, 21.5305}, {169.813, 21.5891}, {169.794, 21.6095}, {169.715, 21.5305},
 {169.734, 21.5101}, {169.813, 21.5891}, {169.715, 21.5305}, {169.813, 21.5891},
 {169.715, 21.5305}, {169.813, 21.5891}, {169.715, 21.5305}, {169.813, 21.5891},
 {169.794, 21.6095}, {169.715, 21.5305}, {169.734, 21.5101}, {169.813, 21.5891},
 {169.715, 21.5305}, {169.813, 21.5891}, {169.794, 21.6095}, {169.794, 21.6095},
 {169.715, 21.5305}, {169.734, 21.5101}, {169.813, 21.5891}, {169.794, 21.6095},
 {169.715, 21.5305}, {169.734, 21.5101}, {169.734, 21.5101}, {169.813, 21.5891},
 {169.794, 21.6095}, {169.715, 21.5305}, {169.813, 21.5891}, {169.794, 21.6095},
 {169.715, 21.5305}, {169.734, 21.5101}, {169.813, 21.5891}, {169.794, 21.6095},
 {169.794, 21.6095}, {169.715, 21.5305}, {169.734, 21.5101}, {169.734, 21.5101}}

demodframe = Sign[Re[ Tom5[[1, 1]] SeqI]] /. {-1 -> 1, 1 -> 0}

{1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0,
 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,
 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1}
```

The first and last bits have been lost in the processing

```
frame

{1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0,
 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1,
 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1}

Length[frame]

72

truncatedframe = frame // Drop[#, 1]& // Drop[#, -1]&

{1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0,
 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,
 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1}

demodframe - truncatedframe

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Tom6 = CDMACoarseSynchroniser[ModOutputUnEncodedFrame3 // Drop[#, 250 4]&,
  Goldseqlist // #[[3]]& , 100, 4]

$Aborted

Tom6[[1, 1]] SeqI // Re

{-157.37, -157.056, -157.056, -157.056, -157.056, -157.056, -157.056, -156.862,
 9.75774, -9.3815, 9.3815, -9.3815, 9.3815, -9.18122, -157.37, -156.862, 9.75774,
 -9.18122, -157.176, 9.75774, -9.3815, 9.18122, 157.37, 156.862, -9.75774, 9.18122,
 157.176, -9.75774, 9.3815, -9.18122, -157.37, -156.862, 9.55746, 157.37, 157.056,
 156.862, -9.75774, 9.18122, 157.37, 156.862, -9.75774, 9.18122, 157.176, -9.75774,
 9.3815, -9.18122, -157.37, -156.862, 9.55746, 157.176, -9.75774, 9.3815, -9.18122,
 -157.176, 9.75774, -9.3815, 9.3815, -9.18122, -157.176, 9.75774, -9.18122,
 -157.176, 9.75774, -9.3815, 9.18122, 157.176, -9.55746, -157.37, -157.056, -157.056}

Tom6[[1, 1]] SeqI // Im

{-10.0454, -10.3815, -10.3815, -10.3815, -10.3815, -10.3815, -10.3815, -10.62,
 -15.3789, 15.6783, -15.6783, 15.6783, -15.6783, 15.9004, -10.0454, -10.62, -15.3789,
 15.9004, -10.2839, -15.3789, 15.6783, -15.9004, 10.0454, 10.62, 15.3789, -15.9004,
 10.2839, 15.3789, -15.6783, 15.9004, -10.0454, -10.62, -15.601, 10.0454, 10.3815,
 10.62, 15.3789, -15.9004, 10.0454, 10.62, 15.3789, -15.9004, 10.2839,
 15.3789, -15.6783, 15.9004, -10.0454, -10.62, -15.601, 10.2839, 15.3789,
 -15.6783, 15.9004, -10.2839, -15.3789, 15.6783, -15.6783, 15.9004, -10.2839,
 -15.3789, 15.9004, -10.2839, -15.3789, 15.6783, -15.9004, 10.2839, 15.601,
 -10.0454, -10.3815, -10.3815}

Tom6[[1, 2]] SeqI // Re

{-169.751, -169.733, -169.733, -169.733, -169.733, -169.733, -169.733, -169.638,
 10.444, -10.425, 10.425, -10.425, 10.425, -9.86944, -169.751, -169.638, 10.444,
 -9.86944, -169.656, 10.444, -10.425, 9.86944, 169.751, 169.638, -10.444, 9.86944,
 169.656, -10.444, 10.425, -9.86944, -169.751, -169.638, 9.88847, 169.751, 169.733,
 169.638, -10.444, 9.86944, 169.751, 169.638, -10.444, 9.86944, 169.656, -10.444,
 10.425, -9.86944, -169.751, -169.638, 9.88847, 169.656, -10.444, 10.425, -9.86944,
 -169.656, 10.444, -10.425, 10.425, -9.86944, -169.656, 10.444, -9.86944,
 -169.656, 10.444, -10.425, 9.86944, 169.656, -9.88847, -169.751, -169.733, -169.733}
```

```
Tom6 = CDMACoarseSynchroniser[ModOutputUnEncodedFrame3 // Drop[#, 250 4]&,
  Goldseqlist // #[[3]]& , 100, 4]

DeModulator[{{10.3815 - 157.056 I, 157.056 + 10.3815 I, -10.3815 + 157.056 I,
   -157.056 - 10.3815 I, 10.3815 - 157.056 I, 157.056 + 10.3815 I, -10.3815 + 157.056 I,
   -156.862 - 10.62 I, 15.601 + 9.55746 I, -157.176 - 10.2839 I, 15.601 + 9.55746 I,
   -157.176 - 10.2839 I, 15.601 + 9.55746 I, -157.37 - 10.0454 I, 10.3815 - 157.056 I,
   156.862 + 10.62 I, -15.601 - 9.55746 I, 157.37 + 10.0454 I, -10.62 + 156.862 I,
   9.55746 - 15.601 I, -10.2839 + 157.176 I, 9.75774 - 15.3789 I, -15.6783 - 9.3815 I,
   -9.18122 + 15.9004 I, 10.2839 - 157.176 I, -9.75774 + 15.3789 I, 15.9004 + 9.18122 I,
   -157.176 - 10.2839 I, 15.601 + 9.55746 I, -157.37 - 10.0454 I, 10.3815 - 157.056 I,
   156.862 + 10.62 I, -15.3789 - 9.75774 I, -9.3815 + 15.6783 I, 15.6783 + 9.3815 I,
   9.18122 - 15.9004 I, -10.2839 + 157.176 I, 9.75774 - 15.3789 I, -15.6783 - 9.3815 I,
   -9.18122 + 15.9004 I, 10.2839 - 157.176 I, -9.75774 + 15.3789 I, 15.9004 + 9.18122 I,
   -157.176 - 10.2839 I, 15.601 + 9.55746 I, -157.37 - 10.0454 I, 10.3815 - 157.056 I,
   156.862 + 10.62 I, -15.3789 - 9.75774 I, -9.18122 + 15.9004 I, 10.2839 - 157.176 I,
   -9.55746 + 15.601 I, 10.0454 - 157.37 I, 156.862 + 10.62 I, -15.601 - 9.55746 I,
   157.176 + 10.2839 I, -15.601 - 9.55746 I, 157.37 + 10.0454 I, -10.62 + 156.862 I,
   9.55746 - 15.601 I, -10.0454 + 157.37 I, -156.862 - 10.62 I, 15.601 + 9.55746 I,
   -157.176 - 10.2839 I, 15.3789 + 9.75774 I, 9.18122 - 15.9004 I, -10.0454 + 157.37 I,
   -157.056 - 10.3815 I, 10.3815 - 157.056 I, 157.056 + 10.3815 I}, {-7.65342 - 169.733 I,
   169.733 - 7.65342 I, 7.65342 + 169.733 I, -169.733 + 7.65342 I, -7.65342 - 169.733 I,
   169.733 - 7.65342 I, 7.65342 + 169.733 I, -169.638 + 6.41574 I, 17.2252 + 9.88847 I,
   -169.656 + 6.44773 I, 17.2252 + 9.88847 I, -169.656 + 6.44773 I, 17.2252 + 9.88847 I,
   -169.751 + 7.68541 I, -7.65342 - 169.733 I, 169.638 - 6.41574 I, -17.2252 - 9.88847 I,
   169.751 - 7.68541 I, 6.41574 + 169.638 I, 9.88847 - 17.2252 I, 6.44773 + 169.656 I,
   10.444 - 16.382 I, -16.412 - 10.425 I, -9.86944 + 17.2553 I, -6.44773 - 169.656 I,
   -10.444 + 16.382 I, 17.2553 + 9.86944 I, -169.656 + 6.44773 I, 17.2252 + 9.88847 I,
   -169.751 + 7.68541 I, -7.65342 - 169.733 I, 169.638 - 6.41574 I, -16.382 - 10.444 I,
   -10.425 + 16.412 I, 16.412 + 10.425 I, 9.86944 - 17.2553 I, 6.44773 + 169.656 I,
   10.444 - 16.382 I, -16.412 - 10.425 I, -9.86944 + 17.2553 I, -6.44773 - 169.656 I,
   -10.444 + 16.382 I, 17.2553 + 9.86944 I, -169.656 + 6.44773 I, 17.2252 + 9.88847 I,
   -169.751 + 7.68541 I, -7.65342 - 169.733 I, 169.638 - 6.41574 I, -16.382 - 10.444 I,
   -9.86944 + 17.2553 I, -6.44773 - 169.656 I, -9.88847 + 17.2252 I,
   -7.68541 - 169.751 I, 169.638 - 6.41574 I, -17.2252 - 9.88847 I,
   169.656 - 6.44773 I, -17.2252 - 9.88847 I, 169.751 - 7.68541 I, 6.41574 + 169.638 I,
   9.88847 - 17.2252 I, 7.68541 + 169.751 I, -169.638 + 6.41574 I, 17.2252 + 9.88847 I,
   -169.656 + 6.44773 I, 16.382 + 10.444 I, 9.86944 - 17.2553 I, 7.68541 + 169.751 I,
   -169.733 + 7.65342 I, -7.65342 - 169.733 I, 169.733 - 7.65342 I}}]

Tom6[[1, 1]] SeqI // Re

{157.056, 157.056, 157.056, 157.056, 157.056, 157.056, 157.056, 156.862, -9.55746,
 -157.176, 9.55746, 157.176, -9.55746, -157.37, -157.056, -156.862, 9.55746, 157.37,
 156.862, -9.55746, -157.176, 9.75774, -9.3815, 9.18122, 157.176, -9.75774, 9.18122,
 157.176, -9.55746, -157.37, -157.056, -156.862, 9.75774, -9.3815, 9.3815, -9.18122,
 -157.176, 9.75774, -9.3815, 9.18122, 157.176, -9.75774, 9.18122, 157.176, -9.55746,
 -157.37, -157.056, -156.862, 9.75774, -9.18122, -157.176, 9.55746, 157.37,
 156.862, -9.55746, -157.176, 9.55746, 157.37, 156.862, -9.55746, -157.37,
 -156.862, 9.55746, 157.176, -9.75774, 9.18122, 157.37, 157.056, 157.056, 157.056}

Tom6[[1, 1]] SeqI // Im

{-10.0454, -10.3815, -10.3815, -10.3815, -10.3815, -10.3815, -10.3815, -10.62,
 -15.3789, 15.6783, -15.6783, 15.6783, -15.6783, 15.9004, -10.0454, -10.62, -15.3789,
 15.9004, -10.2839, -15.3789, 15.6783, -15.9004, 10.0454, 10.62, 15.3789, -15.9004,
 10.2839, 15.3789, -15.6783, 15.9004, -10.0454, -10.62, -15.601, 10.0454, 10.3815,
 10.62, 15.3789, -15.9004, 10.0454, 10.62, 15.3789, -15.9004, 10.2839,
 15.3789, -15.6783, 15.9004, -10.0454, -10.62, -15.601, 10.2839, 15.3789,
 -15.6783, 15.9004, -10.2839, -15.3789, 15.6783, -15.6783, 15.9004, -10.2839,
 -15.3789, 15.9004, -10.2839, -15.3789, 15.6783, -15.9004, 10.2839, 15.601,
 -10.0454, -10.3815, -10.3815}

Tom6[[1, 2]] SeqI // Re

{-169.751, -169.733, -169.733, -169.733, -169.733, -169.733, -169.733, -169.638,
 10.444, -10.425, 10.425, -10.425, 10.425, -9.86944, -169.751, -169.638, 10.444,
 -9.86944, -169.656, 10.444, -10.425, 9.86944, 169.751, 169.638, -10.444, 9.86944,
 169.656, -10.444, 10.425, -9.86944, -169.751, -169.638, 9.88847, 169.751, 169.733,
 169.638, -10.444, 9.86944, 169.751, 169.638, -10.444, 9.86944, 169.656, -10.444,
 10.425, -9.86944, -169.751, -169.638, 9.88847, 169.656, -10.444, 10.425, -9.86944,
 -169.656, 10.444, -10.425, 10.425, -9.86944, -169.656, 10.444, -9.86944,
 -169.656, 10.444, -10.425, 9.86944, 169.656, -9.88847, -169.751, -169.733, -169.733}
```

```
Table[CDMAPositionFinder[ModOutputUnEncodedFrame3 // Drop[#, 250 4 + i 20]&,
    Goldseqlist // #[[3]]&, 4] // Flatten // Abs // Max,
 {i, 251, 350}]
```

```
{22.6392, 22.6392, 22.6392, 17.4452, 17.6558, 17.6558, 17.6558, 17.6558, 17.4676,
 17.4676, 18.088, 18.088, 18.088, 18.088, 19.6342, 19.6342, 19.6342, 19.6342, 19.6342,
 13.457, 16.0791, 16.0791, 16.0791, 16.0791, 19.95, 19.95, 19.95, 19.95, 19.95,
 18.3042, 16.6993, 16.7542, 23.0831, 25.3773, 25.3773, 25.3773, 25.3773, 24.601,
 24.601, 24.601, 18.7221, 20.5255, 26.9209, 26.9209, 26.9209, 26.9209, 18.9705,
 18.9705, 18.9705, 20.3921, 22.6392, 22.6392, 22.6392, 22.6392, 17.4452, 17.6558,
 17.6558, 17.6558, 17.6558, 17.4676, 17.4676, 18.088, 18.088, 18.088, 18.088,
 19.6342, 19.6342, 19.6342, 19.6342, 19.6342, 13.457, 16.0791, 16.0791,
 16.0791, 16.0791, 19.95, 19.95, 19.95, 19.95, 19.95, 18.3042, 16.6993,
 16.7542, 23.0831, 25.3773, 25.3773, 25.3773, 25.3773, 24.601, 24.601, 24.601,
 18.7221, 20.5255, 26.9209, 26.9209, 26.9209, 26.9209, 18.9705, 18.9705, 18.9705}
```

```
Max[%]
```

```
26.9209
```

```
Map[Max, %]
```

```
{17.6558, 17.6558, 17.6558, 17.6558, 17.4676, 17.4676, 18.088, 18.088, 18.088, 18.088,
 19.6342, 19.6342, 19.6342, 19.6342, 19.6342, 13.457, 16.0791, 16.0791,
 16.0791, 16.0791, 19.95, 19.95, 19.95, 19.95, 19.95, 18.3042, 16.6993,
 16.7542, 23.0831, 25.3773, 25.3773, 25.3773, 25.3773, 24.601, 24.601, 24.601,
 18.7221, 20.5255, 26.9209, 26.9209, 26.9209, 26.9209, 18.9705, 18.9705,
 18.9705, 20.3921, 22.6392, 22.6392, 22.6392, 22.6392, 17.4452}
```

References

1 "Binary Sequences with Gold-Like Correlation but Larger Linear Span" by Serdar Boztas and P. Vijay Kumar IEEE Trans. on Information Theory Vol 40 No 2, March 1984

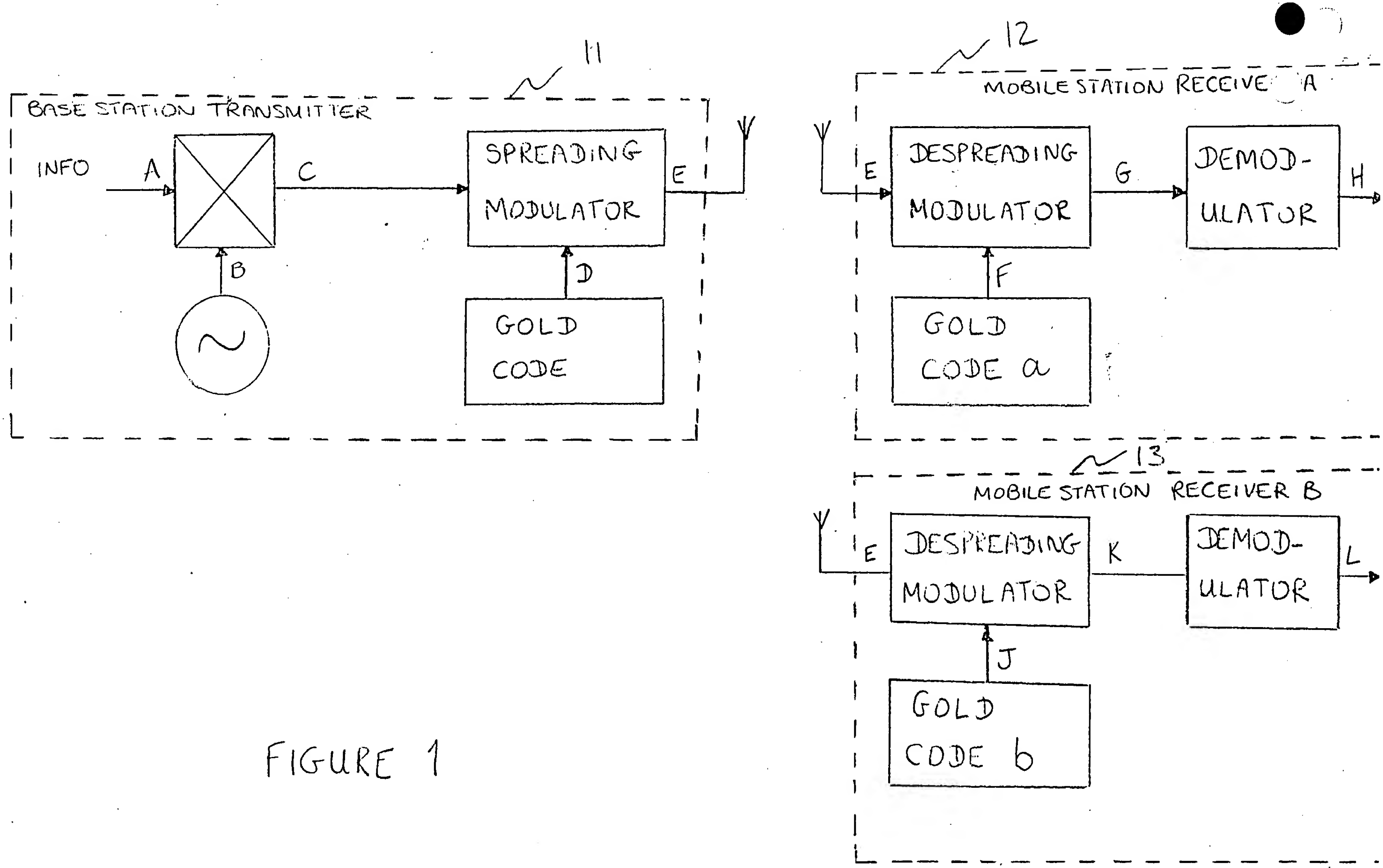

11
BASE STATION TRANSMITTER
INFO
A
B
C
SPREADING MODULATOR
D
GOLD CODE
E
12
MOBILE STATION RECEIVE A
E
DESPREADING MODULATOR
F
GOLD CODE a
G
DEMODULATOR
H
13
MOBILE STATION RECEIVER B
E
DESPREADING MODULATOR
J
GOLD CODE b
K
DEMODULATOR
L

FIGURE 1

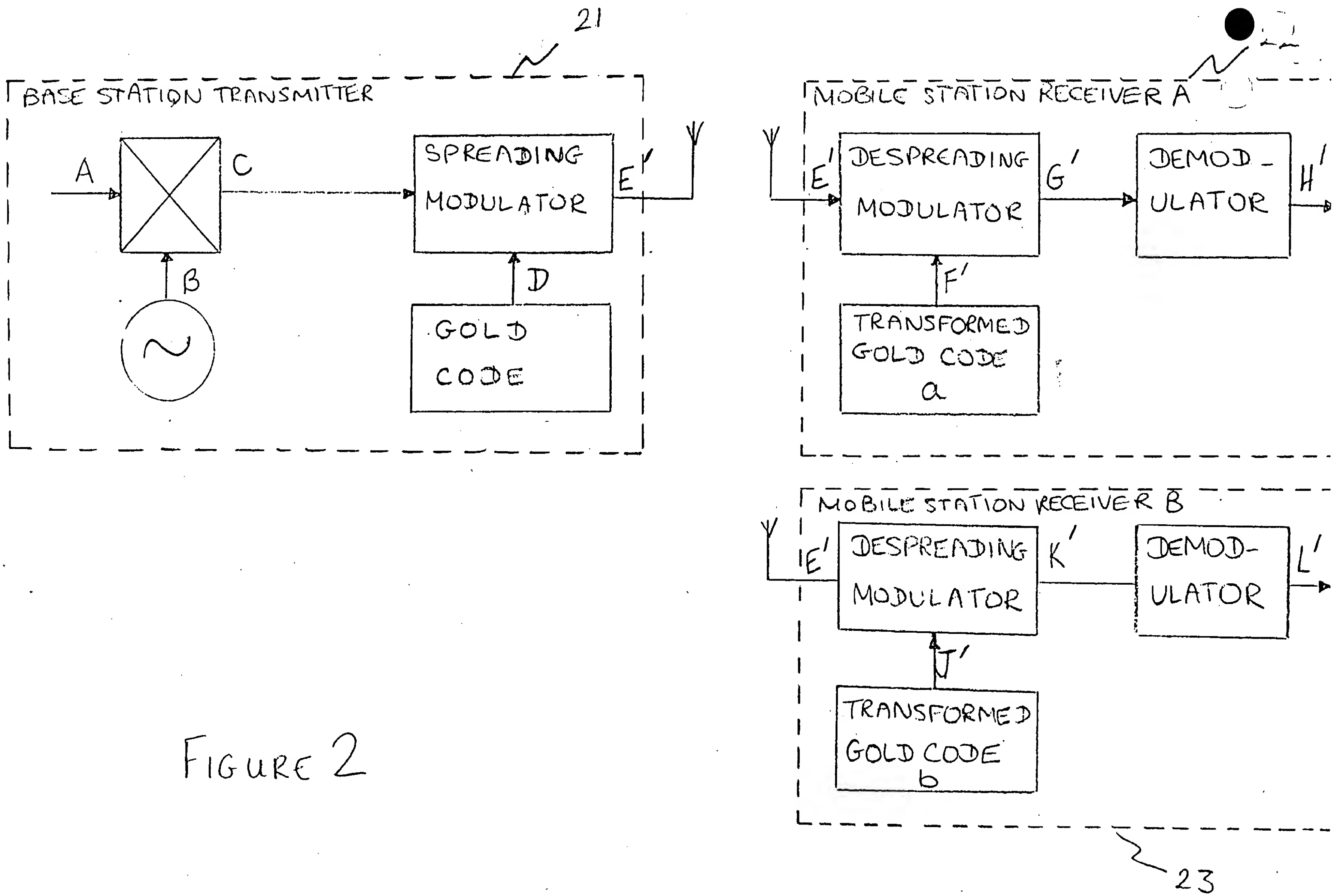

21
BASE STATION TRANSMITTER
A
C
B
SPREADING MODULATOR
E'
D
GOLD CODE
MOBILE STATION RECEIVER A
E'
DESPREADING MODULATOR
G'
DEMOD-ULATOR
H'
F'
TRANSFORMED GOLD CODE a
MOBILE STATION RECEIVER B
E'
DESPREADING MODULATOR
K'
DEMOD-ULATOR
L'
J'
TRANSFORMED GOLD CODE b
23

FIGURE 2

CDMA TRANSMITTER

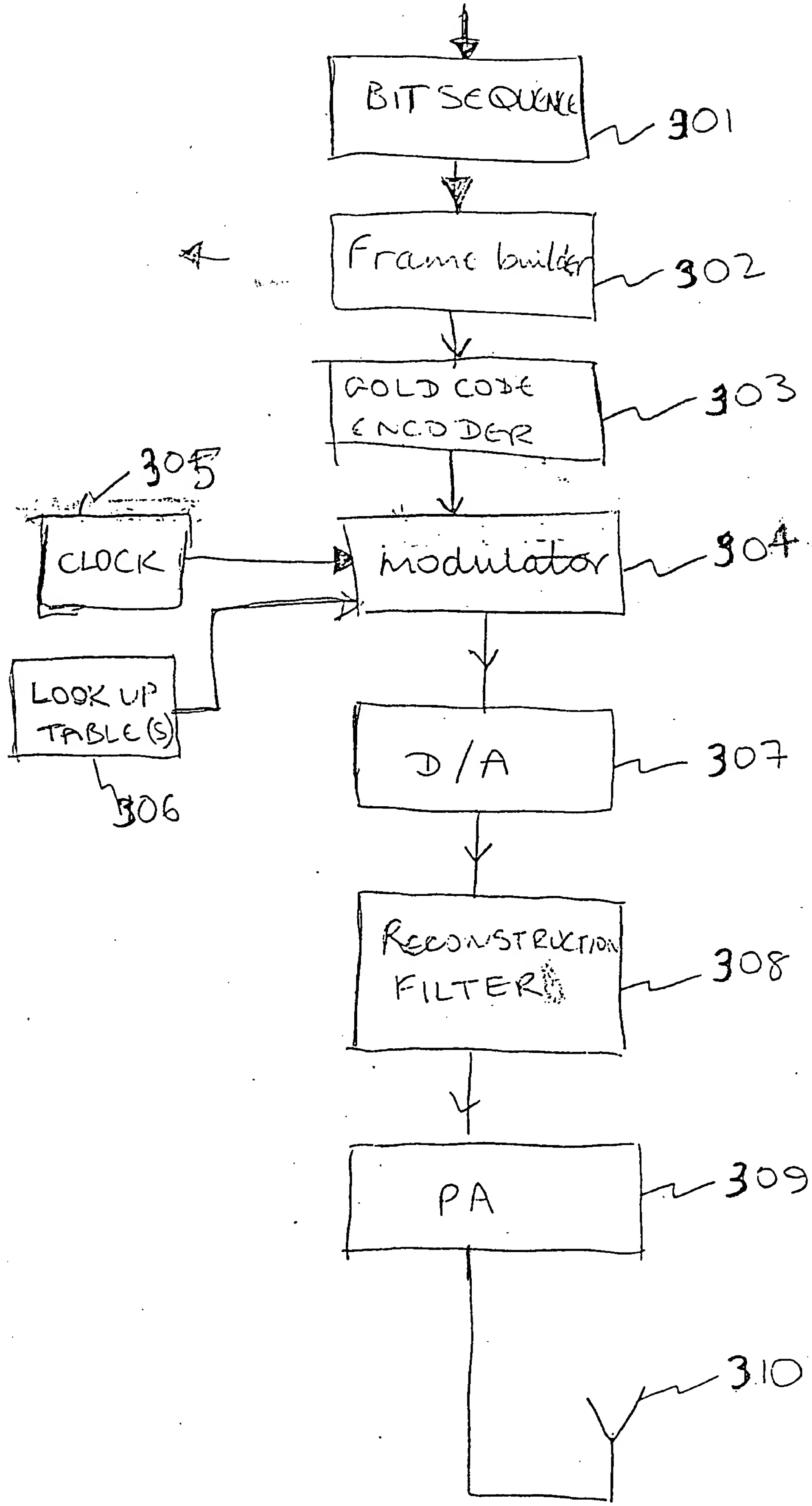


FIGURE 3

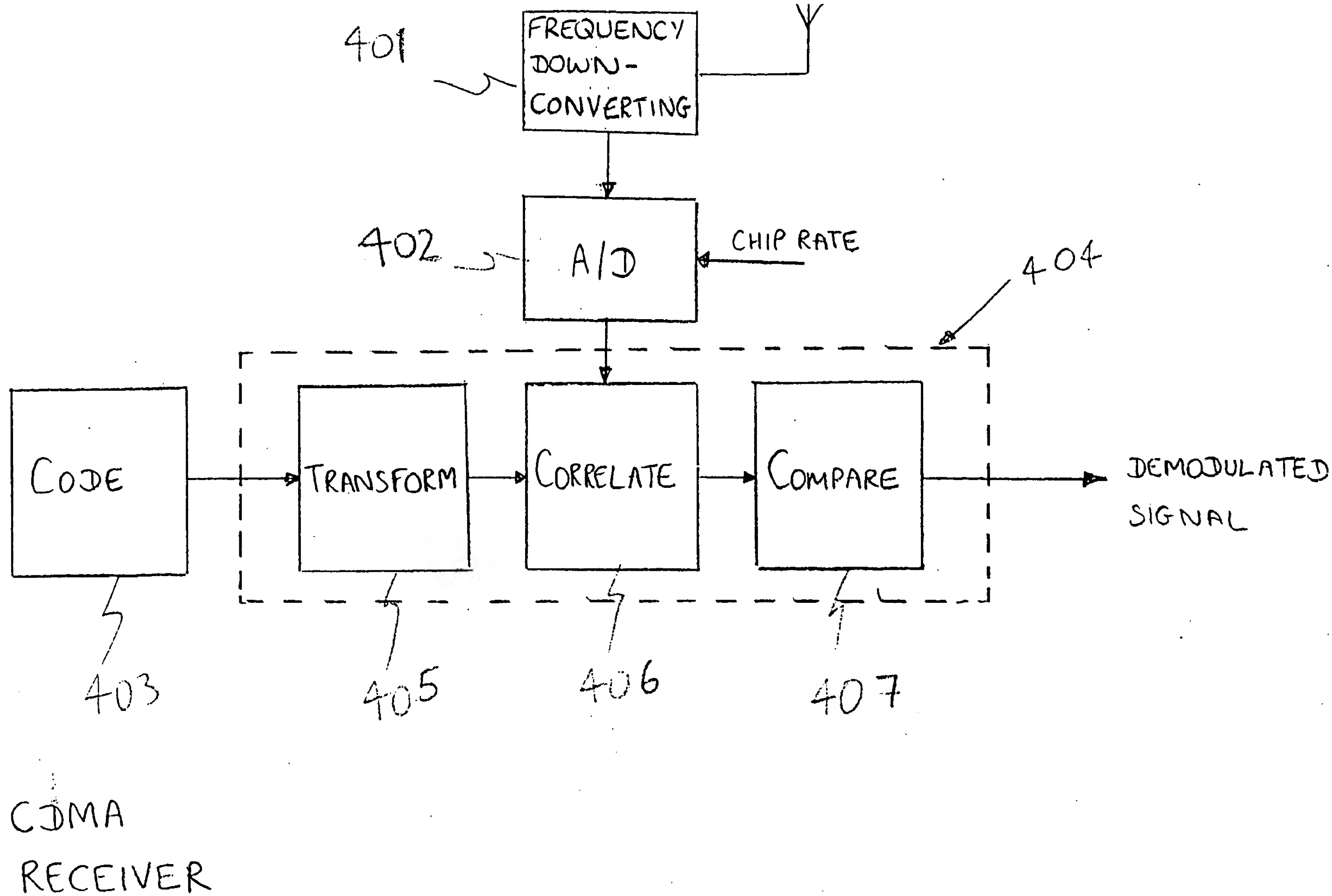
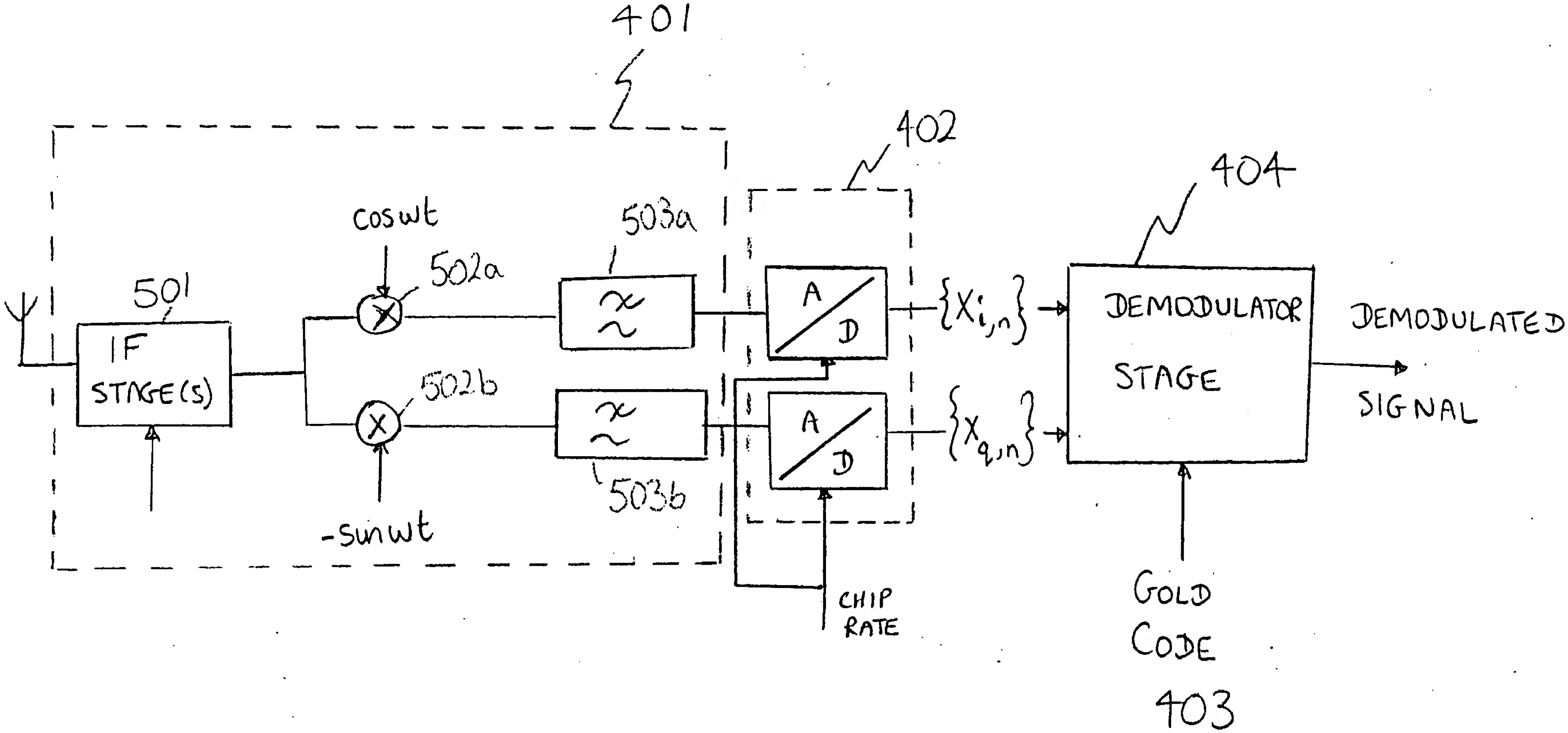

401
FREQUENCY DOWN-CONVERTING
402
A/D
CHIP RATE
404
CODE
TRANSFORM
CORRELATE
COMPARE
DEMODULATED SIGNAL
403
405
406
407
CDMA RECEIVER

FIGURE 4



CDMA RECEIVER

FIGURE 5 (a)

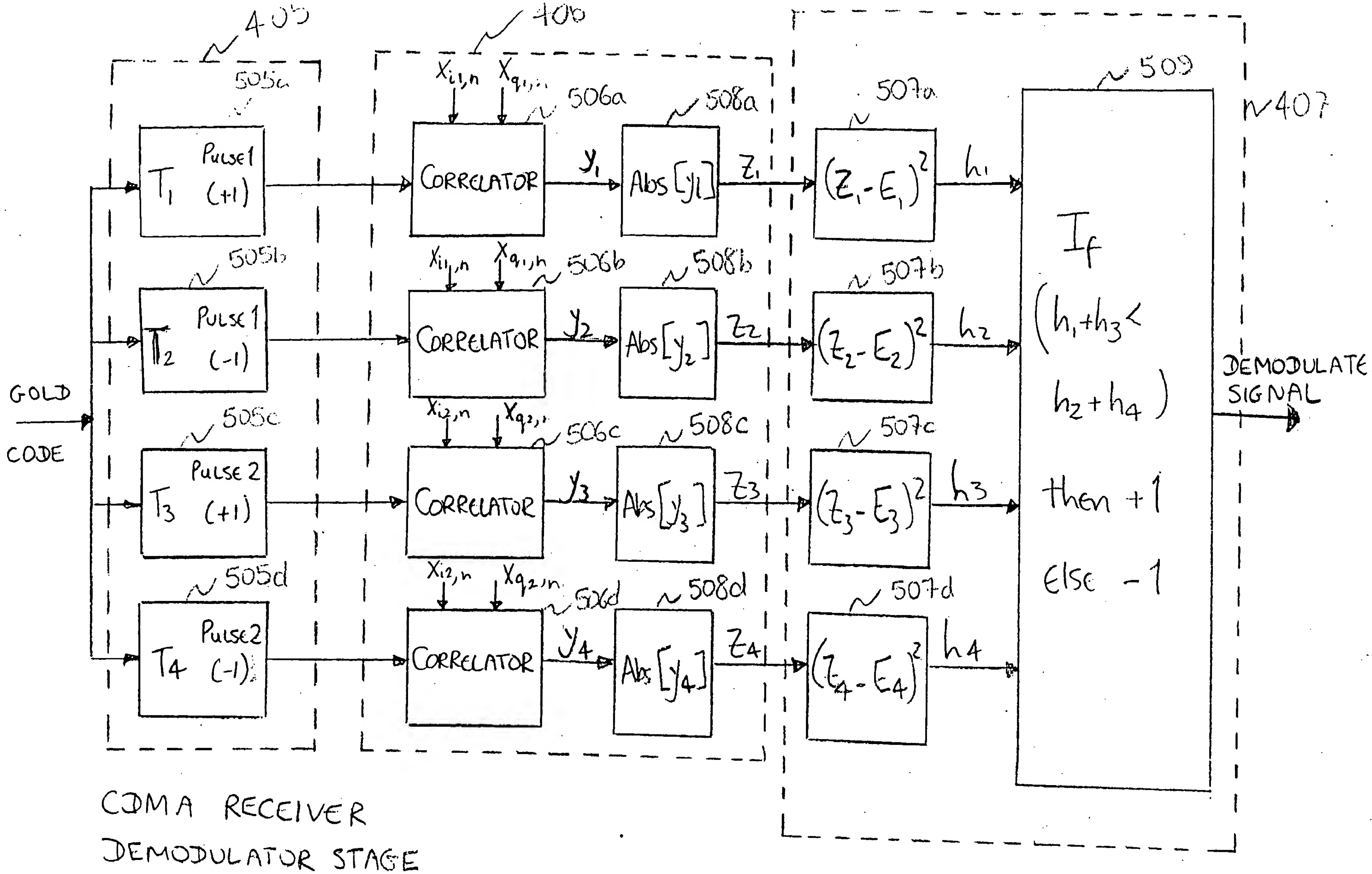
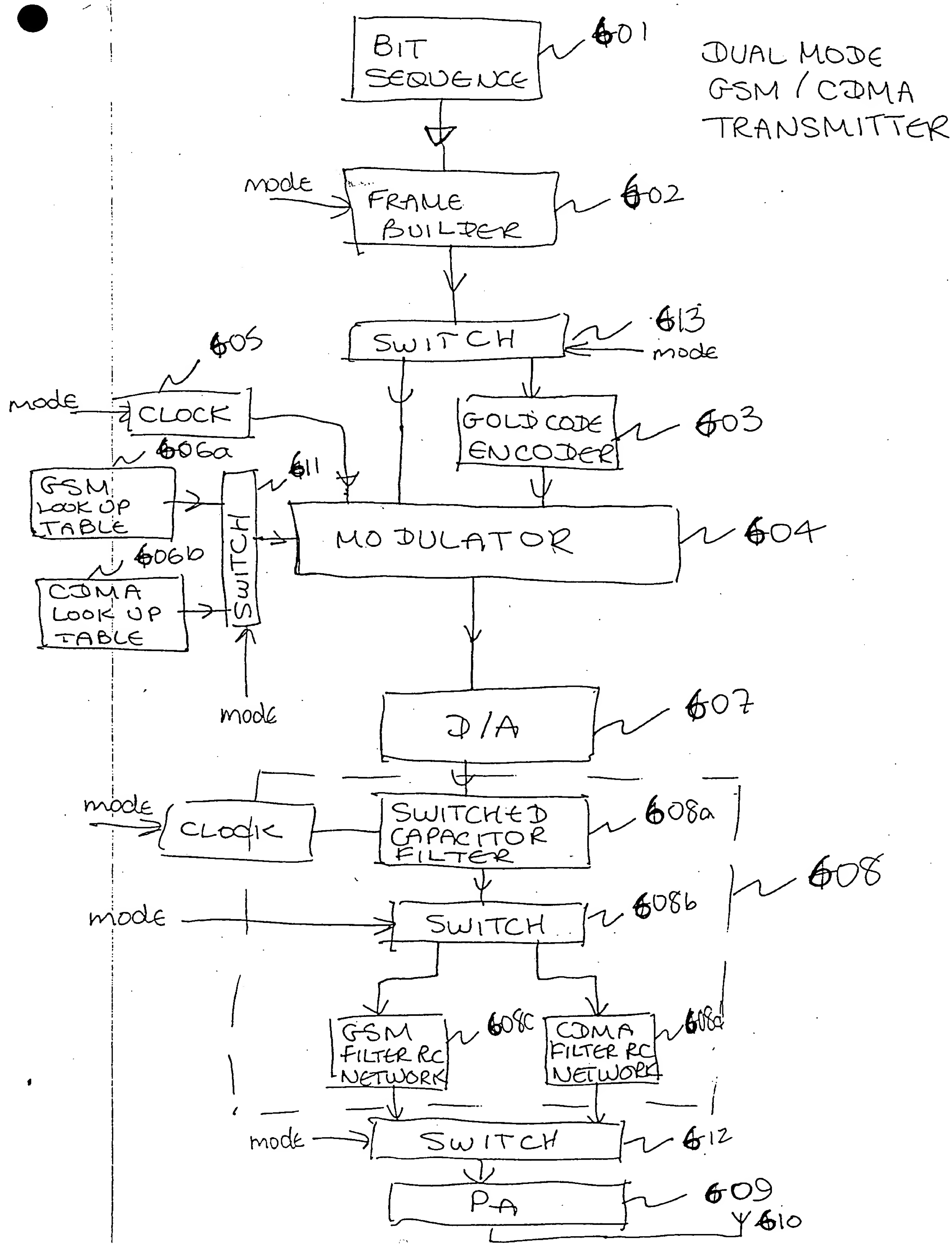

GOLD
CODE
405
505a
505b
505c
505d
Pulse1
T_1 (+1)
Pulse 1
T_2 (-1)
Pulse 2
T_3 (+1)
Pulse2
T_4 (-1)
406
$X_{i1,n}$ $X_{q1,n}$
506a
506b
506c
506d
$X_{i2,n}$ $X_{q2,n}$
CORRELATOR
CORRELATOR
CORRELATOR
CORRELATOR
y_1
y_2
y_3
y_4
508a
508b
508c
508d
Abs[y_1]
Abs[y_2]
Abs[y_3]
Abs[y_4]
z_1
z_2
z_3
z_4
507a
507b
507c
507d
$(z_1 - E_1)^2$
$(z_2 - E_2)^2$
$(z_3 - E_3)^2$
$(z_4 - E_4)^2$
h_1
h_2
h_3
h_4
509
If
$(h_1 + h_3 < h_2 + h_4)$
then +1
Else -1
407
DEMODULATE
SIGNAL
CDMA RECEIVER
DEMODULATOR STAGE

FIGURE 5 (b)

FIGURE 6


DUAL MODE
GSM / CDMA
TRANSMITTER
BIT SEQUENCE
601
mode
FRAME BUILDER
602
SWITCH
613
mode
605
mode
CLOCK
606a
GOLD CODE ENCODER
603
GSM LOOK UP TABLE
611
SWITCH
MODULATOR
604
606b
CDMA LOOK UP TABLE
mode
D/A
607
mode
CLOCK
SWITCHED CAPACITOR FILTER
608a
608
mode
SWITCH
608b
GSM FILTER RC NETWORK
608c
CDMA FILTER RC NETWORK
608d
mode
SWITCH
612
PA
609
610

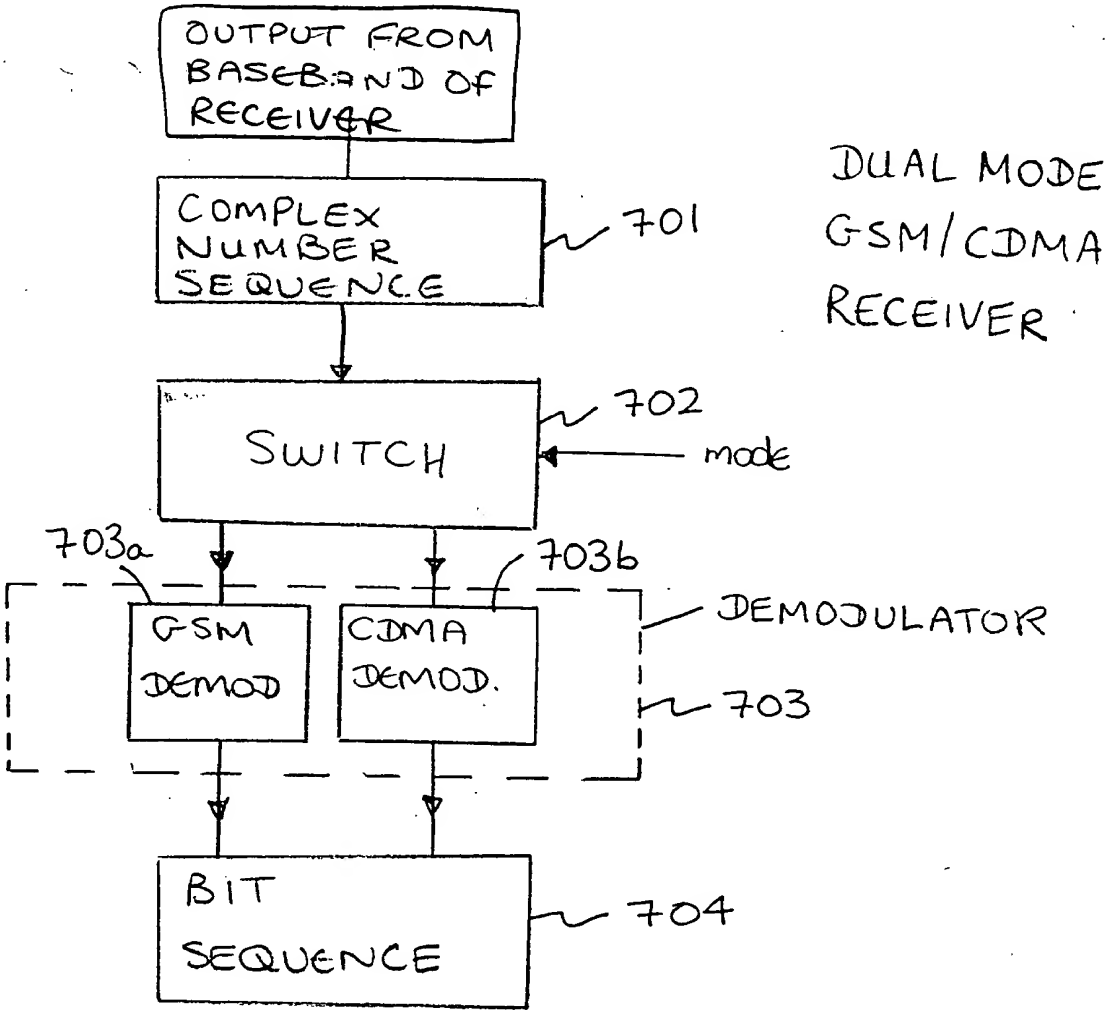

OUTPUT FROM BASEBAND OF RECEIVER
COMPLEX NUMBER SEQUENCE
701
DUAL MODE GSM/CDMA RECEIVER
SWITCH
702
mode
703a
703b
GSM DEMOD
CDMA DEMOD.
DEMODULATOR
703
BIT SEQUENCE
704

FIGURE 7

For a +1 bit

Figure 8(a) – received signal

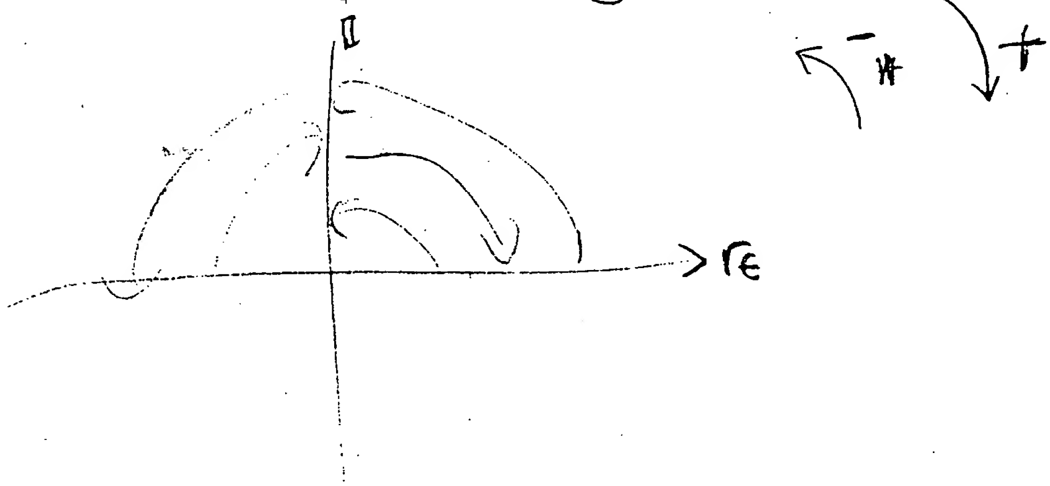
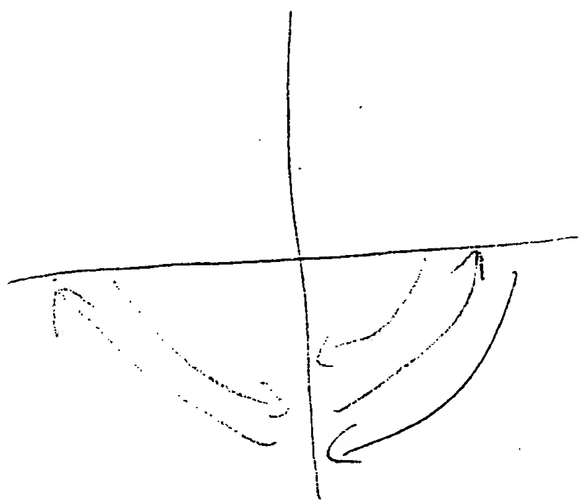


Figure 8(b).



1.0m a — 1 bit

Figure 9a. recd signal

Figure 9b

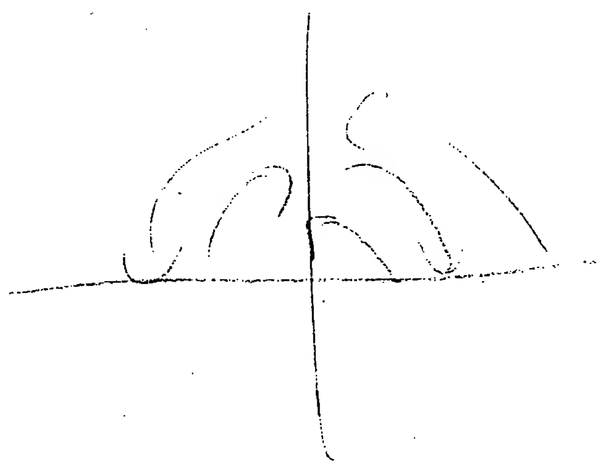


FIGURE 10

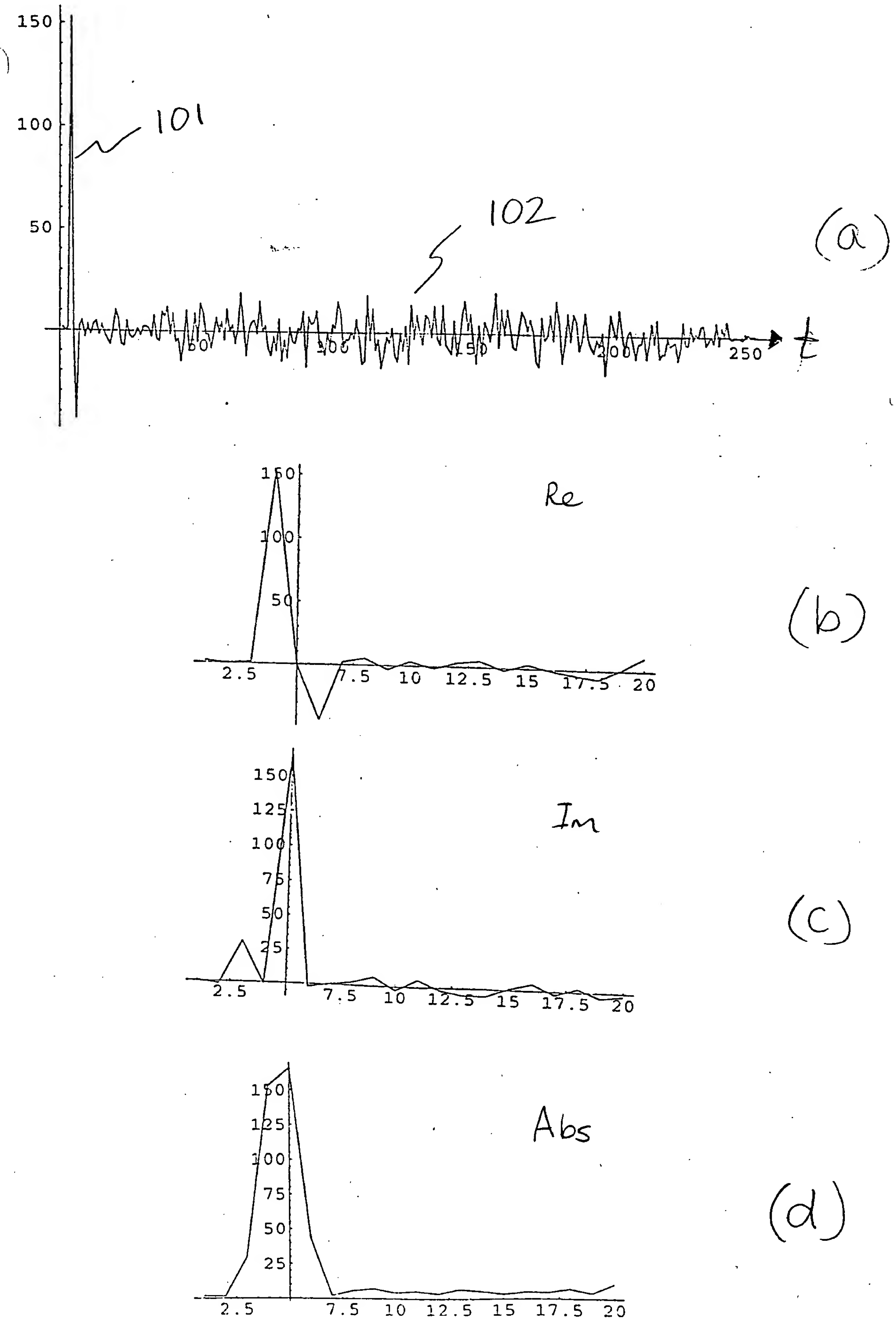

150
100
50
101
102
50
100
150
200
250
t
(a)
150
100
50
2.5
7.5
10
12.5
15
17.5
20
Re
(b)
150
125
100
75
50
25
2.5
7.5
10
12.5
15
17.5
20
Im
(c)
150
125
100
75
50
25
2.5
7.5
10
12.5
15
17.5
20
Abs
(d)

FIGURE 11

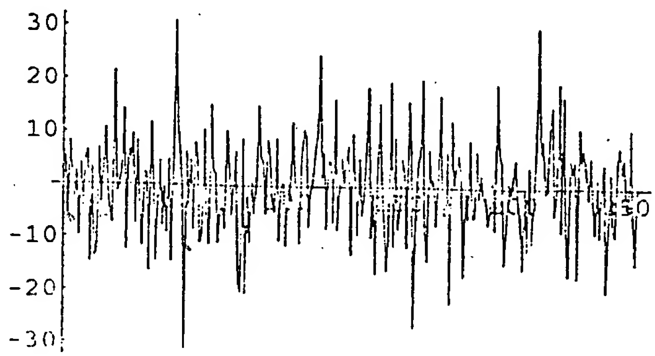


FIGURE 12 (a)

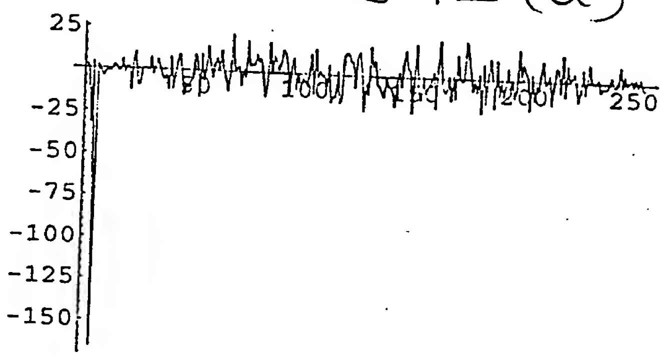


FIGURE 12 b

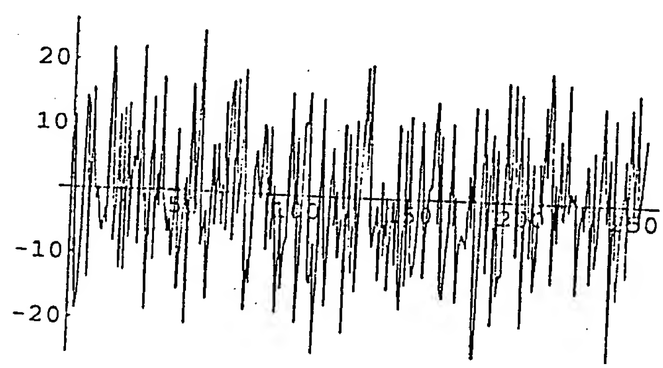
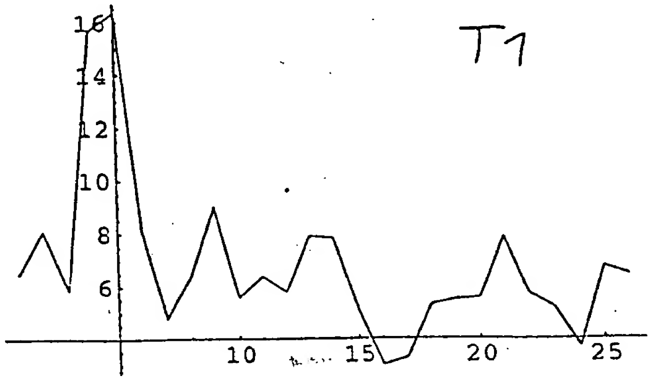
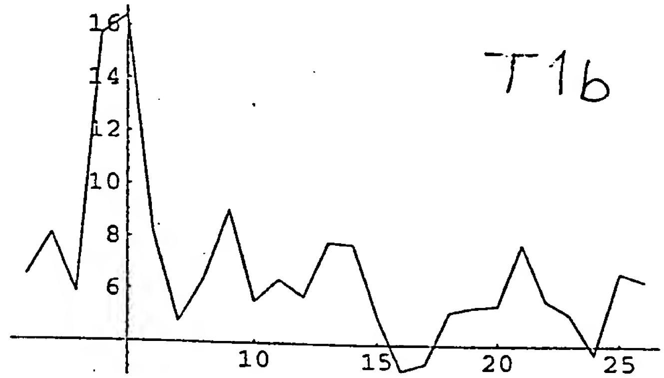


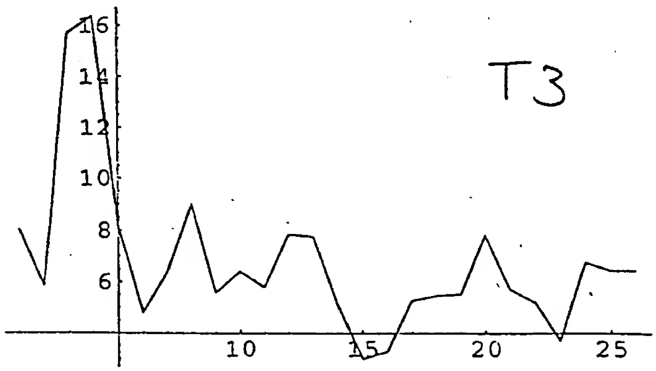
FIGURE 15



(a)

(b)





(c)

(d)